FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Communication Patterns in Project-Oriented Organizations

João André Nogueira Ferreira

Master in Electrical and Computers Engineering

Supervisor: Professor Doutor José António Rodrigues Pereira de Faria

February 20, 2017

Resumo

A comunicação representa uma vertente absolutamente fulcral dentro das organizações orientadas a projeto, seja na gestão diária do trabalho ou nas relações existentes entre os indivíduos que a constituem, existindo um conjunto alargado de ferramentas disponíveis que facilitam esta colaboração. Porém, tamanha variedade pode representar um obstáculo a uma comunicação mais eficaz e eficiente dentro das próprias organizações.

Se por um lado, um bom uso da comunicação interpessoal numa organização pode conduzir a melhorias significativas na produtividade e na obtenção de resultados por parte dos seus colaboradores, uma aplicação inapropriada das ferramentas de comunicação, nomeadamente do *e-mail*, poderá também levar a uma perda de eficiência e disrupção constante no fluxo de trabalho.

Assim, torna-se essencial criar ferramentas pensadas e estruturadas para este tipo de organizações, que suportem todo o fluxo de comunicação entre níveis hierárquicos distintos ou equivalentes, procurando ao máximo padronizar os diferentes tipos de interação existentes.

Nesse sentido, a presente Dissertação aborda esta temática com dois objetivos principais. O primeiro foca-se na identificação de padrões de comunicação existentes entre os colaboradores numa organização orientada a projeto. Tendo por base a análise efetuada a estes fluxos de informação, o segundo objetivo principal centra-se na implementação de um módulo de comunicação que permita suportar os diferentes padrões identificados.

Numa fase inicial da Dissertação foi feita uma familiarização com as ferramentas de desenvolvimento necessárias à elaboração do projeto, assim como uma análise dos requisitos para todo o sistema. Após esta fase, foi elaborado um modelo do domínio que identifica as relações existentes entre diferentes entidades presentes no sistema. Posteriormente a implementação da solução desenvolvida foi realizada utilizando SQL Server 2014 e Visual Studio 2015.

Para a elaboração da solução, foram tidos em consideração conceitos relacionados com a produtividade pessoal e comportamentos no trabalho. Estes, por sua vez, estão intimamente ligados a algumas metodologias como o *Getting Things Done* e *Behavior Design*.

O sistema desenvolvido permite uma troca de mensagens ágil e intuitiva, à semelhança do que acontece com as aplicações de *messaging*, garantindo em simultâneo uma organização equivalente ao *e-mail*, o que permite um armazenamento de dados mais estruturado para posterior uso. Constata-se então que este se trata de um sistema híbrido que permitirá ao utilizador final uma comunicação expedita com os outros. Além deste mecanismo de troca de mensagens, é ainda identificado e explicado um conjunto de padrões de comunicação que definem modelos de interação entre os utilizadores. Estes são claramente orientados a determinada ação, seja ela validação, aceitação, colocação de questão, entre outros. De uma forma geral, é expectável que o fluxo de comunicação existente numa organização seja simplificado e simultaneamente mais conciso, evitando redundâncias e perdas de informação ao longo do tempo, com a utilização deste sistema.

Além de comprovar a validade de conceitos explorados na Dissertação, nomeadamente da gestão do trabalho, o módulo de gestão da comunicação desenvolvido permitiu testar e validar as

funcionalidades principais, através da sua integração na plataforma de gestão de trabalho empresarial uONE, desenvolvida pela empresa Dreamo.

Abstract

Communication represents an essential component of an organization way of work, whether it is in the daily work management or in the existent interactions among individuals that are part of it. A vast range of tools are available for this purpose. However, it can represent a disturbance inside an organization, as its diversity leads to discordant means of communication.

In one hand, a proper handling of interpersonal communication inside an Organization can improve significantly its productivity and the results of its employees. On the other hand, an unsuitable application of communication tools, namely *e-mail*, can lead to inefficiency and to a continuous disruption in the workflow.

Thus, the development of structured and planned tools for this purpose is essential, as it will support all the communication flow between distinct or alike hierarchical levels, standardizing different kinds of existent interactions.

With this purpose, the present Dissertation approaches the problem with two main objectives. The first concerns the identification of common communication patterns between collaborators, in a Project-Oriented Organization (POO). Based on this analysis, the second objective focus on the implementation of a module that is able to support the distinct communication patterns identified.

In an initial stage of the project, a familiarization with the development technology was needed, as well as the specification of the system requirements. After that, a domain model comprehending all the fundamental relationships between entities was elaborated. This stage was followed by the implementation of the solution using SQL Server 2014 and Visual Studio 2015.

Several concepts regarding personal productivity and behavior at work were taken into consideration to elaborate the final solution. These are closely related to methodologies as *Getting Things Done* (GTD) and *Persuasive Behavior Design* (PBD).

The developed system allows an agile and intuitive message exchanging, similar to messaging applications, assuring simultaneously a structure comparable to *e-mail*, which permits a more organized data storage, and in consequence an easier traceability of information. Therefore, this system might be considered a hybrid between those two channels of communication, providing an expeditious communication with others. Moreover, a set of communication patterns, which define interaction models among users, is identified and explained. These patterns are undoubtedly oriented to explicit actions, like validation, confirmation, asking and others. Thus, it is expected that, in general, the communication flow inside an organization is simplified and at the same time more concise, avoiding redundancies and information losses over time.

Besides proving the validity of the explored concepts, regarding the work management, during this Dissertation, the module developed for communication management allowed the test and validation of its main functionalities, already as an integrated part of the enterprise work management platform uONE, developed by Dreamo.

Acknowledgments

Em primeiro lugar queria agradecer ao Professor José António Faria pelo apoio e disponibilidade demonstrados ao longo de todo este período. Sem dúvida que o seu enorme entusiasmo pelo tema fez com que eu me sentisse muito mais motivado para este projeto.

Também é de realçar a preciosa colaboração prestada tanto pelo Helder Marques, como pelo João Silva, membros da Dreamo. Por isso e pela amizade que criei com eles, estou-lhes bastante grato.

Em todo este processo, muitas outras pessoas tiveram um papel fundamental e, portanto, não será demais mencionar algumas delas. Ao João Almeida, ao Ruben Dias e ao Manuel Silva, gostava de deixar uma mensagem de profundo agradecimento pelo apoio prestado em diversos momentos, mas mais que isso pela excelente amizade que temos e que espero conservar por muitos e bons anos. Deixo também um agradecimento especial à Catarina Costa por me ter apoiado constantemente durante esta jornada.

Gostava ainda de deixar uma palavra de apreço aos meus familiares mais próximos, que sempre me proporcionaram as condições necessárias para o meu desenvolvimento pessoal e académico. Aos meus pais, também agradeço por serem um exemplo perfeito de perseverança, humildade e dedicação, incutindo em mim esses mesmos valores ao longo de toda a minha vida.

João André Ferreira

"The man who moves a mountain begins by carrying away small stones."

Confucius

Contents

Resumo i

Abstract iii

1 Introduction 1
- 1.1 Context . . . 1
- 1.2 Motivation . . . 2
- 1.3 Objectives . . . 3
- 1.4 Development Methodology . . . 3
- 1.5 Document Structure . . . 4

2 State of the Art 5
- 2.1 Context . . . 5
- 2.2 Patterns and Behaviors in Communication . . . 7
  - 2.2.1 Dependency on E-mail . . . 8
  - 2.2.2 Getting Things Done . . . 8
- 2.3 Commercial Solutions . . . 9
  - 2.3.1 Enterprise Project Management Software . . . 9
  - 2.3.2 Collaboration Tools . . . 11

3 System Requirements and Conceptual Model 13
- 3.1 Requirements . . . 13
- 3.2 Conceptual Model . . . 15

4 Proposed Solution 17
- 4.1 Overall Analysis . . . 17
- 4.2 Communication Patterns . . . 18
- 4.3 Communication Organizer . . . 26
- 4.4 Use Cases Matrix . . . 28

5 Communication Organizer Implementation 31
- 5.1 Development Technology and Tools . . . 31
- 5.2 Application Architecture . . . 32
- 5.3 Database Architecture . . . 32
  - 5.3.1 Relational Model . . . 33
  - 5.3.2 Communication Organizer Entities . . . 33
- 5.4 Inbox . . . 34
  - 5.4.1 Talk . . . 36
  - 5.4.2 Quick Reply . . . 37

5.4.3 Nudge . . . 38
5.5 Search . . . 38
5.5.1 Talks Overview . . . 38
5.5.2 Integration with Work Items . . . 39

6 System Testing and Validation **41**
6.1 Talk Flow Testing . . . 41
6.2 Technical Integration . . . 46
6.3 uONE Team Validation . . . 46
6.4 Independent End-User Validation . . . 47

7 Conclusions and Future Work **49**
7.1 Future Work . . . 50

References **51**

List of Figures

2.1 The three main management dimensions inside an organization . . . 6
2.2 The three main dimensions for communication activities . . . 7
2.3 Wrike's communication module . . . 9
2.4 Share comments on any item in EPM Live [1] . . . 10
2.5 Slack's communication module . . . 11
2.6 Share files and collaborate with people in Yammer [2] . . . 12

3.1 System conceptual model . . . 15

4.1 Message properties . . . 17
4.2 Communication patterns . . . 18
4.3 Messaging Pattern from Sender's View . . . 19
4.4 Messaging Pattern from Recipient's View . . . 20
4.5 Nudge Pattern . . . 21
4.6 Assignment Pattern . . . 22
4.7 Invitation Pattern . . . 23
4.8 Validation Pattern . . . 23
4.9 Notification Pattern . . . 24
4.10 Reporting Pattern . . . 25
4.11 Schema of inbox page . . . 26
4.12 Archiving Mechanism Modelling . . . 27
4.13 Talk properties . . . 28

5.1 Application architecture schema . . . 32
5.2 Relational Model . . . 33
5.3 Personal Organizer . . . 35
5.4 Controls used to implement the drag and drop functionality . . . 36
5.5 Form that starts a new talk . . . 36
5.6 Presentation of messages associated to a talk inside a popup . . . 37
5.7 Dropdown button to select a project and the remaining form details . . . 37
5.8 Form for quick reply . . . 37
5.9 Nudge action . . . 38
5.10 Presentation of Search page . . . 39
5.11 Generic function that allows to expand or collapse the rows . . . 39
5.12 Presentation of messages associated to a talk inside a project's page . . . 40

6.1 Starting a new talk named "Talk Test" . . . 41
6.2 Message sent in the sender Pending's box . . . 42
6.3 Message shown in the recipient's inbox page . . . 42

6.4 Creating a new message in the Talk form 42
6.5 Message sent updated in the sender Pending's section 43
6.6 Creating a new message in the Quick Reply form 43
6.7 Sending an instantaneous message 43
6.8 Receiving a new message 44
6.9 Message read in Later's box 44
6.10 Drag and Drop between the two central containers 44
6.11 Replying to the message received 45
6.12 Archiving action triggered 45
6.13 Searching for a conversation 45
6.14 Nudging other user 46
6.15 Presenting the nudged message 46

List of Tables

4.1 Talk Management Use Cases 29

5.1 Programming Tools used in implementation 32

Abbreviations

CSS	Cascading Style Sheets
GTD	Getting Things Done
HTML	HyperText Markup Language
JS	Javascript
PBD	Persuasive Behavior Design
POO	Project-Oriented Organization
SQL	Structured Query Language
WBA	Web-Based Application
WI	Work Item

Chapter 1

Introduction

This chapter presents and explains the context, the motivation and the objectives defined for this Master Dissertation. It is also presented the methodology used for its accomplishment, as well as the structure defined for the present document.

1.1 Context

One of information technology's purpose is simplifying the production and distribution of information, but most of the times it is inappropriately used, or somehow its potential is limited.

To achieve this objective, organizations tend to significantly invest in complex platforms to support the daily work of their teams, many times without relevant success. Multiple factors can explain the negative results of the implementation of such tools in project management, such as the excess of complexity according to the level of knowledge of the users and the inherent inertia from workers to change [3] and adapt to new tools.

In the past, project management was essentially used to accomplish requirements presented by a customer. Teams, typically gathered in a single location [4], would develop a solution specifically for that client, allocating all the resources available to the project development. Due to business globalization and technology advancements, new styles of projects have arisen where people and organizations have to work and cooperate all over the world [5].

Therefore, the organizations have been seeking for new technologies and software to help them evolve and be more efficient in their everyday work. Achieving this is not a straightforward task, taking into consideration the multiple pressures and motives that can lead to a poor or inefficient handling of the tasks and all the information associated.

Typically, in POOs concerning engineering, consulting, or others, operations derive from projects in the form of tasks that presumably will ensure that specific objectives for each project are reached.

Inside an organization, communication is a key factor in the relations established within the departments, teams and with clients. Regarding the different types of interactions, it is clear

that most of them use communication channels such as *e-mail*, *instant messaging*, web-based applications and telephone [6].

Concerning the use of *e-mail*, is clear that it has become a main way of communication and information exchange, and this occurs in almost every organization.

However, the excess of *e-mail* exchange and the diversity of the communication channels leads to losses of relevant information details, as users cannot filter and give equal importance to all data. The efficiency is also affected significantly by the recurrent search for information through the multiple conversations established. Throughout time, this lack of information traceability [7] will end up being noticed in most commonly used collaboration tools .

In response, many companies' and researchers' efforts are involved in a continuous improvement of the communication process, which hopefully will lead to new tools and solutions able to improve the daily work in POO.

1.2 Motivation

In this context, a spin-off from FEUP entitled as Dreamo – Balanced Work Systems has been developing solutions for several organizations with the purpose of improving the quality of the work accomplished, knowledge sharing and the collaboration inside these companies.

As referred in section 1.1, organizations have been using different tools to manage work. Often, these applications do not cover all the dimensions associated to the work management, leading to systems that are not fully integrated, in both vertical and horizontal flows of information. Besides, these platforms usually focus on planning tasks, generating Gantt Charts, in the assignment of tasks and documents management. The collaborative component, regarding the interactions among team members, is frequently ignored, or just basic modules are implemented where instant and isolated messages can be exchanged.

Using these systems might be enough for small companies or ones with limited resources, where the users are closer to each other and there is not a considerable number of projects ongoing. However, in organizations with a considerable size and multiple projects running simultaneously, this type of software solution may not be enough. In fact, these mechanisms cause a high level of inefficiency to the whole organization and create delays along the workflow, planning and reporting process.

In the matter of communication process, the majority of organizations tend to use individual and independent tools such as Microsoft Outlook or browsers to have access to e-mail, instead of using an application that is fully integrated in their platforms.

Thus, Dreamo pretends to offer its clients an integrated system with tools that allows users to keep information permanently trackable even after projects, tickets, tasks and others are finished, reducing the amount of time spent searching for e-mails and logs of communication.

In a POO, due to the quantity of stages that must be accomplished, the information produced and developed, and later reported to project managers, should be available at any time during the development phase, but especially for future work development.

Moreover, generating a constant rhythm in the process of communication is important to assure better results from team members. Achieving this means to create mechanisms capable of avoiding constant interruptions, that otherwise would certainly lead to a decreasing focus and consecutively a lower efficiency [8]. Thus, the development of patterns capable of establishing rules in communication is necessary for increasing the productivity inside the organizations.

1.3 Objectives

This Master Dissertation project contemplates two main objectives, the analysis and the identification of interaction patterns that can be used to standardize the communication flows existing within a POO, and the development of a module for communication management purposes, providing the necessary tools to accomplish a productive and simple system to the end-user.

The communication module should cover several aspects regarding work management, archiving system, permissions management and information traceability.

As this solution will be developed in partnership with Dreamo, it is expected that the module can be implemented in the platform for enterprise work management already being developed, uONE.

Finally, the system should be prepared for future changes, regarding new features and enhancements needed to assure the best user experience as well as continuous improvement.

1.4 Development Methodology

This dissertation project follows a system engineering approach, contemplating a reliable guideline for analysis and development in this kind of processes. The employed methodology establishes distinct stages that should be followed sequentially:

- Analyze the state of the art concerning the areas of interest to the project;
- Analyze and specify the final system requirements;
- Formulate the system conceptual model;
- Propose a solution for the identified problem;
- Implement and integrate the developed system;
- Test and validate the employed solution;
- Document the implemented solution.

Taking this development model approach into consideration, it is manifest that this dissertation project contemplates several tasks regarding the proper development and achievement of the expected outcomes. The obtained results, throughout the distinct stages, will be fully integrated as a system to be implemented in the platform (uONE) already existent.

The first step focus essentially on the requirements analysis followed by the specifications for the system to be implemented. This has a particular relevance, as the specifications represent the base for designing and implementing the whole system. Also, the conceptual model will establish the relationships present in the system, being later transposed to the final database.

Then, the simulation of interfaces, based on mockups, is developed for representation of the final appearance of the application module.

Having the architecture of the system fully designed, implementation takes a relevant role in the development of this dissertation. During this stage, regarding the priorities defined and the usability expected, specific features from the overall solution will be developed. There are several features implemented such as messaging, archiving and management mechanisms, and overall information searching, allowing the testing and validation of the main objectives previously defined for this dissertation.

All these iterative steps of implementation, testing and consequent validation will be recurrent and extensive, to ensure a successful integration of the developed modules. Using this approach, the communication module integration with the existing work planning and management platform will be more coherent and least prone to errors.

In the final stage of implementation process, the developed solution was tested with one Dreamo's client in order to validate the module of communication management and its utility to the target client.

1.5 Document Structure

This document is composed by seven chapters. In the present chapter 1, the scope and motivation for this dissertation are fully described, as well as the expected final result and the methodology used during the development process. Chapter 2 contemplates the State of the Art. In chapter 3, the requirements for this dissertation are specified, and the model and concepts associated to the system developed are presented and explained. Then, in chapter 4, the aim is to outline and justify the main choices regarding the application module development, describing and fully detailing the proposed solution regarding the identified and categorized communication patterns, and the communication module that will support them as the engine of the whole system. Chapter 5 provides information related to implementation matters, regarding the communication organizer, archiving and management mechanisms, as well as organization of the database, by presenting illustrative parts of the system implemented. After that, in chapter 6, typical situations to test and validate the system settled are demonstrated.

Finally, in chapter 7, the conclusions taken from this dissertation project development are exhibited, as well as possible future enhancements to enrich this communication module.

Chapter 2

State of the Art

This chapter presents a panorama on the subject discussed in this dissertation. Section 2.1 will introduce the application scope of this research. After that, an introduction to patterns and behaviors in communication, as well as some techniques to be more productive are presented in section 2.2.

Ultimately, a set of commercial solutions available in the market are identified, regarding the communication management perspective, in section 2.3.

2.1 Context

Encouraged by the technological advances during the past decades, the environment in organizations has become more globalized, with people communicating constantly using different means and not necessarily on the same place [4]. Also, the information shared may have different destinations, inside or outside the organizations, such as team members, clients and suppliers.

In POOs, the relationships between those entities are tight, since the purpose of these companies is to develop products and services that fit distinct requirements, in strict cooperation with clients [9].

As Raymond and Bergeron stated [10], the management of these projects and its success is highly correlated with the resources and the project management tools available for the teams.

Project management is a field in constant evolution and continuous improvement, promoting the best results within organizations, departments, and specially in teams. The United States Project Management Institute states that the purpose of project management is nothing more than meeting project requirements previously established, using the knowledge, skills and tools necessary to achieve it. Also, it defines project as a temporary effort, by one or more individuals, to create a product or service, tangible or not, within a deadline [11].

Nowadays, multidisciplinary teams are a reality present in a vast majority of POOs, such as consulting and engineering firms. The engagement between different fields of knowledge is necessary for a proper achievement of established requirements [12]. This led to change the traditional perspective on project management.

By the year of 1995, Helbrough [13] stated that computer-assisted collaboration could reduce the amount of time spent in meetings, with more accurate data and easier to interpret and analyze. The capacity of immediate distribution and availability of produced documentation for further editing or processing could also be reinforced.

Schwalbe [14] presents the concept of enterprise project management software as a tool that is capable of integrating information across an organization, from multiple projects within the past, present and future.

Associated with this type of software, three distinct dimensions must be considered as shown in Figure 2.1.

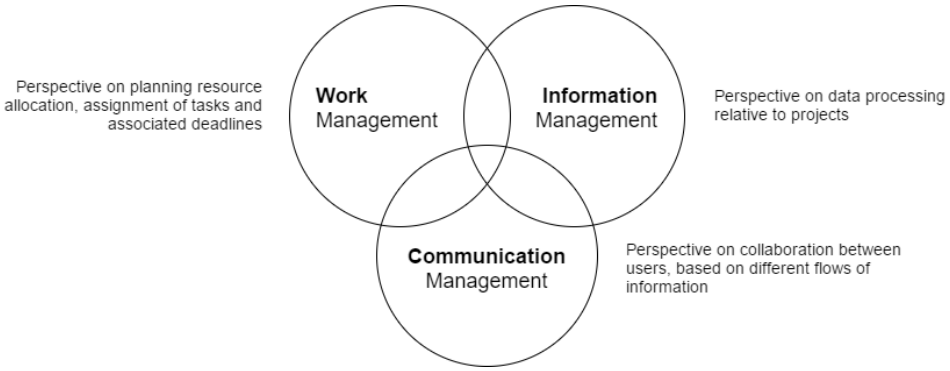

Figure 2.1: The three main management dimensions inside an organization

Improving the features of software concerning communication management is important to aid and enhance communications within the teams. However, the most difficult and challenging part in an organization is to motivate and prepare people to change habits and learn new techniques for communicating. Only with a good plan for communication management this can be assured [6].

Gareis [15] stated that the use of appropriate information and communications technology infrastructures is absolutely relevant to guarantee the project management on a professional level. Being so, assuring a proper plan and collaboration software, capable of sustaining all communications, represents a challenge for every organization.

Communication among team members and stakeholders has a major role in the development of the projects, as referred in [16]. Regarding the project communications management, there are three main processes to consider. One refers to the communications management planning that intends to develop an approach to obtain a plan to meet needs and requirements according to the resources available. The other two are communications management and control that contemplates the mechanisms used for management of the information produced and collected, and ensures that communications throughout the entire project life cycle are correctly used, respectively.

As Qian and Zhen-hua [17] affirm, that an effective communication is the base of a project, since it is the basis for information exchange.

The actors involved in this system have specific functions such as project managers, team members and stakeholders. Based on the communication processes already mentioned, distinct

dimensions for communication activities are established by the Project Management Institute [16], such as:

- Internal interactions associated to projects, and external relationships with other projects, organizations or stakeholders;
- Formal and informal communications such as reports and discussions, respectively;
- Vertical interactions regarding communications between different management levels inside the organization. Horizontal collaboration for operational purposes;
- Written and oral information.

Badiru [18] also presents several dimensions for communication activities within organizations, as presented in Figure 2.2.

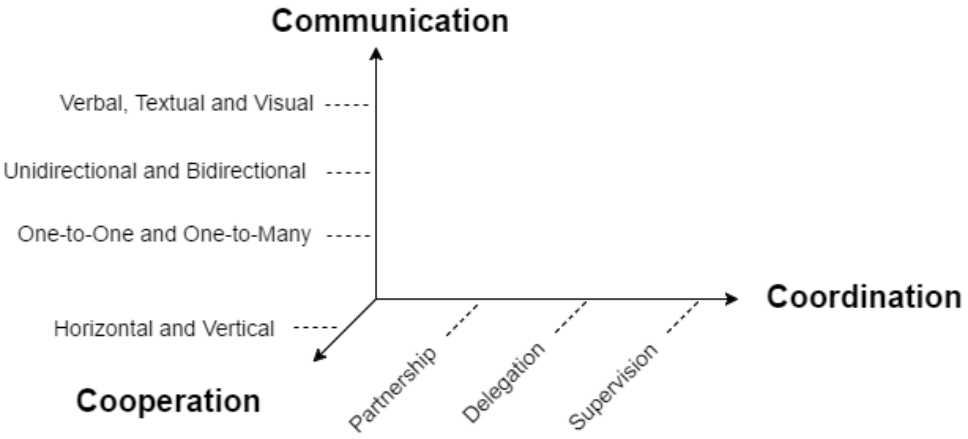

Figure 2.2: The three main dimensions for communication activities

2.2 Patterns and Behaviors in Communication

About the design of cooperative work, Winograd [19] illustrated an action oriented conversation in a state transition network, and presented a solution for computer-based systems using patterns of interaction that could be found inside organizations, as a request or an offer for example, and other kinds of action that automatically match as an answer to the template used. Those action oriented patterns and highly connected to the context of the conversations, intended to provide a set of templates capable of standardizing and facilitating the communication in work groups.

Fogg [20] presents his behavior model for persuasive design in technology, using three main factors as basis of this study. Supposing a behavior is expected for a specific situation, triggers must be used to instigate the person to behave on that same way it was intended. But a trigger is completely useless if there is lack of motivation or capability. These are both highly important to

assure so that the right stimulation leads to the expected behavior. The motivation will affect the availability of people to do something, while the ability is directly related with the design of the application and its simplicity. An application that does not force the users to learn too much things or that avoids an excessive number of steps to do something will most certainly help them to be capable and susceptible to use it. Obviously, that is the real and complex challenge in technologies development.

2.2.1 Dependency on E-mail

The use of *e-mail* is a primary necessity in all kinds of organizations, but frequently the overload of information arriving can trigger several hidden issues that teams tend to have difficulty dealing with. Constant desktop notifications are sent to provide quick and relevant information to people, but the disruption of focus caused in the receiver isn't, in most cases, not taken into account by the software applications developers [21].

In the study developed by Jackson *et al.* in [22], regarding the price to pay for e-mail interruption, it is demonstrated that the majority of employees tend to check incoming e-mail notifications as they arrive instead of focusing on the work in hands.

The authors of this study research, propose a few advices to handle these interruptions, such as:

- adjust the level of intrusion in the e-mail application by changing the style of notification for alerting when a new e-mail is received;
- reduce the amount of e-mails sent as reply-to-all messages within the company. This requires some training to better understand in which situations should be applied or not;
- display the first three lines of each message to monitor the activity;
- extend the time for updating the mailbox to assure longer times for task focus;
- establish rules to aid the categorization and prioritization for incoming e-mails without requiring immediate attention.

2.2.2 Getting Things Done

The proposed methodology by David Allen [8] in his book, Getting Things Done, intends to provide people a set of mechanisms that help managing their everyday work using several techniques. From those methods, four key ideas have special interest for the topics approached on this master dissertation.

Keeping the inbox empty, with an agile management of the incoming messages, and defining a specific folder for messages that still not have a suitable reply are highly recommended, since it allows people to have full control of what is on their hands. Besides those two ideas, Allen also foments the necessity of defining priorities in every task that must be accomplished. By

doing this, people will be more likely to accept the fact that they cannot do everything, and some prioritization has to be done to guarantee that major goals are achieved. Another key idea is that whatever system is used, it should fit and be easily handled by a person, so that he can be efficient in solving his issues.

2.3 Commercial Solutions

Currently, there are several available commercial solutions in the market for Project Management with collaboration support. Most of these are WBAs, with device and desktop applications also available, used for teams of distinct fields and contexts, such as education, engineering, consulting, manufacturing, construction or design. These systems intend to provide different and capable solutions that can be applied in different contexts regarding work, information and communication management.

Within enterprise project management software and collaboration tools, several solutions could be presented. However, most of them do not have a differentiator aspect that highlights them.

2.3.1 Enterprise Project Management Software

Wrike [23], presented in Figure 2.3, is an application for real-time work management that contemplates the three dimensions of project management software, already referred in the section 2.1.

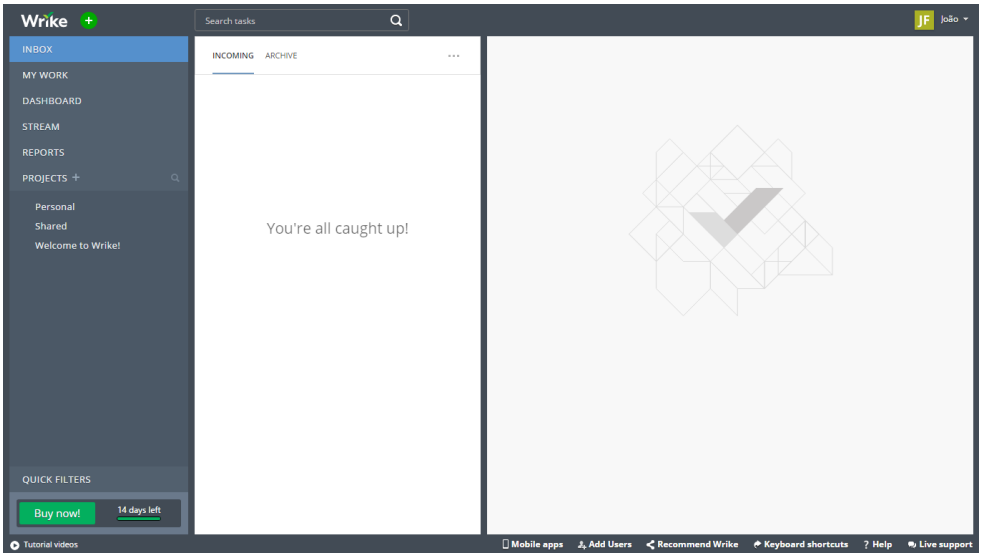

Figure 2.3: Wrike's communication module

This solution is a cloud-based tool that empowers teams to achieve high efficiency even if people are in different places. Regarding collaboration management, this tool presents a dashboard with distinct information for users. Taking in consideration the different projects in progress, it presents the tasks' states for current or archived ones, as well as a live activity stream, where the

communication between team members is done. This is the main communication channel in this application, and it guarantees that everyone is able to share documents as well.

EPM Live [24], shown in Figure 2.4, describes its solution as a fully integrated enterprise work engine that allows teams to manage and collaborate on projects across the organization.

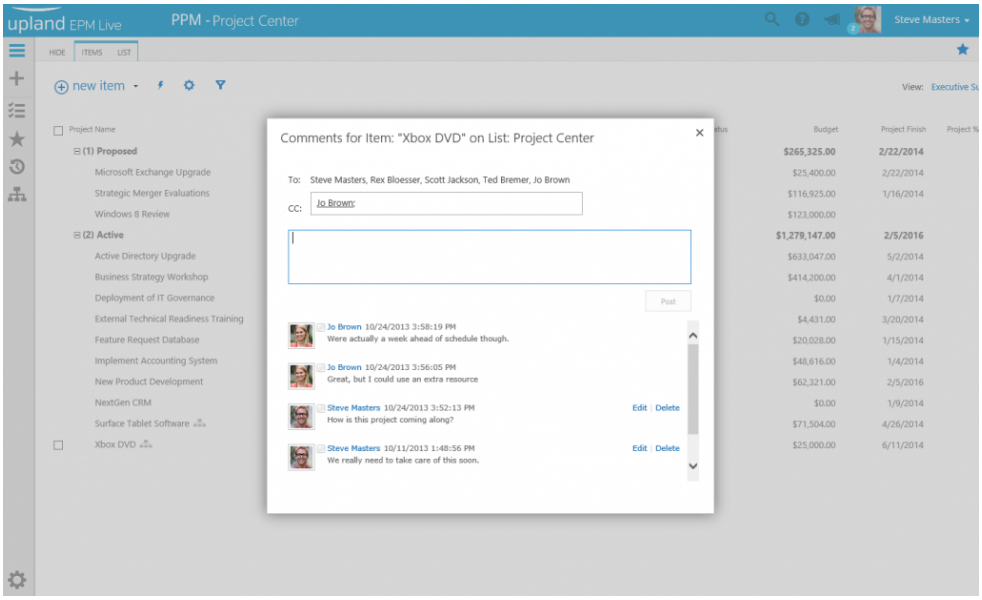

Figure 2.4: Share comments on any item in EPM Live [1]

This software promotes knowledge sharing and better collaboration within the teams to increase productivity through distinct modes of communication. Based on Microsoft SharePoint, this tool establishes the connections with applications such as Microsoft Outlook, to simplify conversations.

Discussion boards related to projects are contemplated in this solution to share information and discuss topics in real-time. These boards are integrated with Microsoft Outlook, allowing the user to reply and post information as preferred.

Automatic reminders based on shared calendars are also proposed, as well as notifications for both individuals and teams regarding project updates and changes [25].

uONE is a WBA offered by Dreamo with a considerable set of functionalities regarding communication, work and information management. This enterprise present its product as their flagship and an innovative tool capable of increasing significantly the efficiency and effectiveness in their clients' work routines.

This platform not only supports projects management, but also another kind of work types as tickets or work orders, being adaptable to several scenarios. By doing this, the system can deal unusual situations that are particular of each client.

Regarding the different dimensions of management already mentioned, uONE platform has been developed, since the new generation of this platform is currently an ongoing project. By the

end of this master dissertation, the communication module in uONE was capable of supporting communication mechanisms in the clients using the new generation of the platform.

2.3.2 Collaboration Tools

Slack [26], displayed in Figure 2.5, presents a quite effective cloud-based solution specifically regarding communication management, based on a new kind of messaging for teams. Claiming a 48.6% reduction in internal e-mail, this application bases its offer in "channels" for organization of team conversations, that can be related to groups, projects or departments.

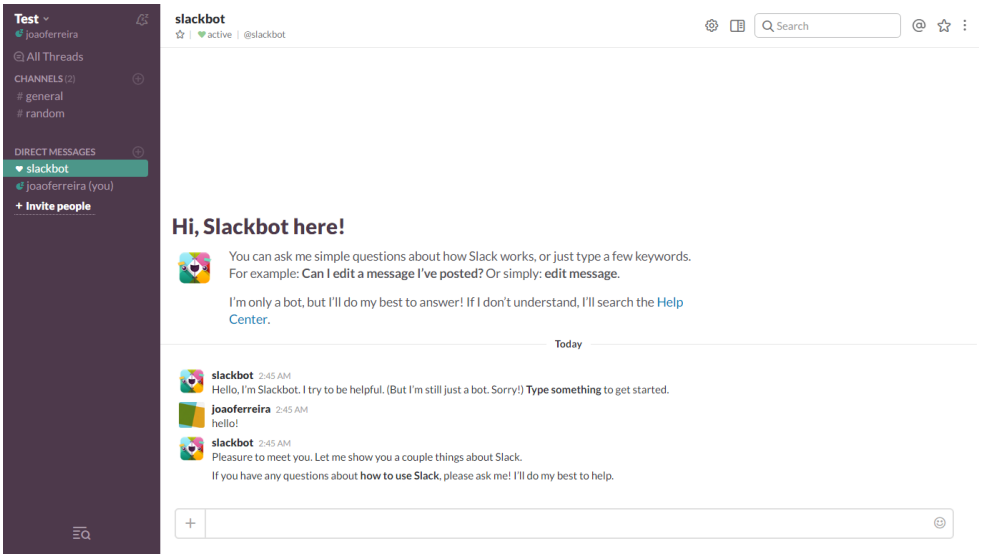

Figure 2.5: Slack's communication module

Communications within the application are divided in two categories:

- **Public channels** — open to anyone, within the team, to see what is going on and join the conversation;
- **Private channels** — only people invited are able to join the conversation and access to confidential information.

Direct messaging is one feature available, where private conversations from one person to another or to multiple receivers (groups) are possible.

This application offers a mechanism of knowledge sharing, using a drag and drop feature, where distinct types of files can be shared within the team. The user is also free to make comments on documents.

System notifications, associated to new messages, are also included in the platforms used by Slack, as well as the search feature that allows users to find information on talks and inside the contents shared, by indexation.

Slack's aim is to bring all the communications together in one place, looking for more productive, transparent and efficient way of working, with no added e-mail.

Microsoft rivals in this market with a comparable tool called Yammer [27], exhibited in Figure 2.6. It follows similar principles as social networks, but within the perspective of the organizations.

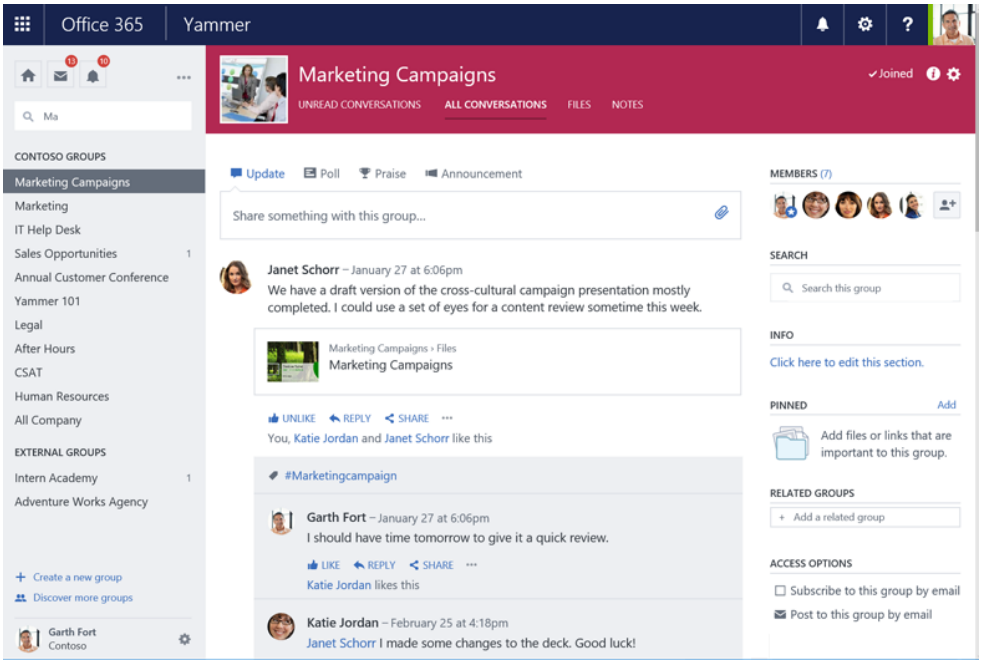

Figure 2.6: Share files and collaborate with people in Yammer [2]

The concept of group is applied as the base of this application, promoting internal and external interactions concerning daily routines of work. The groups allow the users to discuss the projects on going, knowledge sharing with Office 365 fully integrated [28].

Both internal and external collaboration can be assured using this WBA, establishing deeper relationships through group conversations with partners, customers, vendors and members of the organization.

Searching for topics along the information shared is also a feature contemplated in this software, as well as an Inbox, where the user may find all existent conversations. This system, as many others, has integrated notifications.

Chapter 3

System Requirements and Conceptual Model

This chapter contemplates an identification of the system requirements, followed by a conceptualization of the system proposed and with the described implementation in the following chapters.

3.1 Requirements

Dreamo has been developing software with the strong purpose of increasing the added value of characteristic processes in POOs. Considering organizations with a substantial number of employees, the focus is essentially in creating an integrated system that contemplates all the dimensions of project management, mentioned before in chapter 2, in order to obtain a standard platform, entitled as uONE. An important aspect of this solution, that was considered, is the balance between the use of custom options, such as the communication standards, and the flexibility and easy customization according to the client's will or needs. The system developed should be conceived as an extension of the organization in cause, reducing to the minimum possible situations where the client's necessities are limited by this technology.

As the system should have a robust database structure that supports the necessary parametrization to turn this customization from end-user possible, it is expected that different aspects and scenarios are prepared, generalizing as much as possible the proposed system.

Based on multiple received inputs from Dreamo members, it was possible to settle down specific requirements. Thus, the system should provide the following functionalities:

- **Messaging System**
  - Create new conversations;
  - Associate one or multiple users to a new conversation;
  - Specify a subject in a new conversation;
  - Apply a normal or high priority to new created messages;

  - Define to new messages a deadline date as a limit to reply;
  - Attribute one work item to a conversation, maximum;
  - Add new recipients to the conversation, even after a conversation is created;
  - Answer quickly to a received message without requiring to access the integral conversation;
  - Inform if a sent message is read by the recipients or not;
  - Provide to the recipient information about the deadline aimed by the sender;
  - Separate messages in three distinct areas: new ones, messages already read but still needing to reply and the messages sent.

- **Archiving System**
  - Archive messages without restrictions, at any time;
  - Allow the creator of a conversation to archive it whenever he feels that no more relevant informant will come from it;
  - Reopen an archived conversation when a new associated message is sent;

- **Information Traceability**
  - Keep all the records of created conversations;
  - Link a conversation to an existing work item, at any moment;
  - Find messages concerning a specific work item;
  - Search conversations associated to a user;
  - Have access to archived and active conversations, at any instant;

- **Permissions Management**
  - Allow users to follow or unfollow conversations from multiple work items;
  - Be able to define the privacy of a conversation as public or private;
  - Have access to previous messages when added to a private conversation already existent;
  - Provide an open system that grants access to all messages exchanged if a user is member inside a specific conversation or the it is public.

In addition to these specific requirements, the model is also expected to be entirely compatible with uONE platform along with providing great usability for the users, performing accurately and successfully the required activities for their communication management.

3.2 Conceptual Model

The conceptual model represented in Figure 3.1 provides an embracing view of how the developed system will be interrelated, as it establishes the main entities present in the system and the objects that they contain.

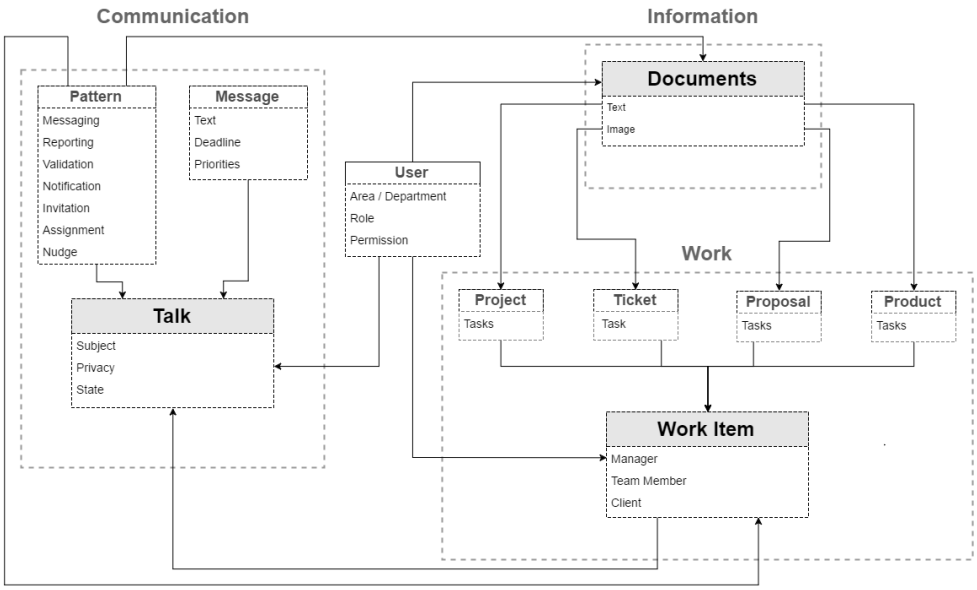

Figure 3.1: System conceptual model

All dimensions regarding an application for project management are present in the same model, grouped in three modules: communication, work and information. As mentioned, this master dissertation focus essentially on the communication module, and for that reason, it is particularly more detailed than the remaining ones.

In order to clarify some expressions used during the present document, a few ideas must be explained for a better understanding of the concepts here presented. Exploring the work's module, the term Work Item (WI) pops out as a main element. A WI is considered to be any element regarding the work module. Thus, the WI is a generalization of different types of work within an organization. In the conceptual model shown, four types of work are identified. However, their purpose is only to exemplify and for that reason, others could be considered. The project represents a process with different stages contemplating a set of tasks to accomplish the established objectives. Tickets may be considered a major task that should be handled and solved as an independent item. The proposal typically refers to commercial affairs that contemplate multiple tasks associated to the elaboration of a proposed solution to a specific client. Product represents the process of development, with a group of tasks that result in a final output. Tasks are considered internal WIs within the previously referred elements.

Regarding the information's module, documents represent resources with distinct types such as images or textual information mostly related to WIs existent in the system.

The last main exhibited entity regards the whole communication management system designed to suit the requirements and the objectives of this master dissertation. Naturally, the solution proposed to support this module is entirely compatible with the remaining management blocks of the platform, establishing direct relationships with them. Talk is the term used to characterize existent conversations between users, with the purpose of creating a dominant entity that aggregates and links every object associated to the communication dimension. From now on, the concept talk will be always used as reference to the presented entity.

The communication pattern concept regards common activities that may be considered as normal and quite standard between individuals, in their interactions. Analyzing these activities, several patterns are distinguishable in a POO.

The messaging pattern represents a situation where two or more individuals exchange ad hoc messages about a specific affair, starting a new talk. Within this pattern, a particular behavior was identified, concerning the usual alerts from someone expecting an answer of the other part. To standardize these type of ad hoc messages, a new pattern was created and named as nudge.

Reporting is another very common activity inside POOs that usually aggregates information about a set of WIs running or completed. Typically, this kind of interaction occurs between two levels of the hierarchical structure, as a member of a team should report to his supervisor, while the latter informs directly the administration.

Approving produced or updated documentation, is another highly normal interaction that may be designated as the validation pattern.

In this context, a notification may also be considered as a pattern. An example of this would be a situation where a specific sender intends to broadcast information to one or multiple recipients, not necessarily expecting an answer from them. This kind of situation is employed regularly to inform members of an organization about multiple affairs such as a new seminar coming, an official announcement and information likely to be relevant for a WI.

Another common pattern identified in POOs, for example, is the invitation for meetings or events. For instance, if a new seminar is being prepared, it is highly recommended that every person likely to be interested is invited. This is of major importance when an organization has a significant number of employees.

Assigning WIs and tasks to collaborators may also be considered as a quite usual and easily standardized procedure in the organization' routines since it requires very specific answers, like accepting or declining an assignment.

As a fundamental part in the entire system, users are an independent entity that link and sustain all the others with continuous stimulus and contents, using the connections provided by the platform. Irrevocably, the projected resultant system is molded to facilitate the end-user's life and keep him stimulated and motivated to use it in his routines. However, the relationships above exhibited will certainly become more evident throughout the document.

Chapter 4

Proposed Solution

This chapter presents the proposed solution for the problem underlying this master dissertation project. Firstly, a context for the solution is exposed, followed by an integral description of how the messaging system is projected.

Finally, specific use cases are presented for a better understanding of the main expected functionalities.

4.1 Overall Analysis

Taking into consideration the project requirements, mentioned in Chapter 3, the proposed solution includes a module for communication management that integrates with Dreamo's uONE platform, and assures the necessary mechanisms to achieve an easy and agile handling of requests, with a structured and complex data background, reflecting techniques for boosting productivity and traceability of information, while increasing the end-user's motivation to collaborate, as much as possible, on the developed platform.

As demonstrated in the conceptual model, presented in section 3.1, the message derives from the Talk entity, representing a sequential but independent item originated by an action of a user as a stimulus to other(s). This object consists of two key attributes with a specific purpose, regarding priority and expectation of a limit date to reply, as shown in Figure 4.1.

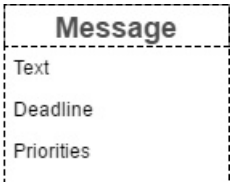

Figure 4.1: Message properties

Prioritizing a message has a single purpose which is to define if it should be categorized as something important to do quickly or if it has an importance considered normal. However, this categorization might be applied from the sender's or recipient's perspective.

The priority can be divided in two exclusive levels, normal or high importance. Besides this, the priority can also be applied in two different contexts, as the sender will assign a level of importance that makes sense to him, as well as the recipient will decide whether to keep that attribute to the received message. This dual attribution of importance to messages will be detailed in section 4.3. Defining a target date to accomplish tasks can also be applied to the message developed system, using it as a reminder of something that should be handled within the proposed time. The idea is to create some sort of pressure on the recipient's side without forcing him to actually settle the response within the stipulated time.

Talks are an integrated component within a system with several WIs that might be in progress, suspended or complete. There may exist established relationships between talks and these items or not, but the association is always possible. This kind of association will be explored later.

4.2 Communication Patterns

Achieving the expected integrated system, that reliably supports the communication flow inside a POO, requires an in-depth view of how the system should be adapted to people and their functions. The purpose of each person within an organization delineate the kind of expected activities for that function. Thus, having tools designed specifically to accomplish those tasks requires an identification of interaction patterns for each type of user.

Thereafter, in the projected patterns, displayed in Figure 4.2, a division between sender and recipient was used to facilitate their comprehension. By doing this, it is intended a clear and objective view of each perspective within the interaction flow.

Figure 4.2: Communication patterns

- **Messaging Pattern**

The pattern exhibited below, in Figure 4.3 and Figure 4.4, describes the distinct stages of a talk, considering a common situation of conversation between users using the proposed system. Messaging is undoubtedly the most pertinent projected pattern, as it characterizes a typical scenario that can be applied within other patterns, extending its use to a wide-ranging application across the system.

Although a conversation is a finite activity, people persistently tend to unconsciously, or not, ignore this and keep it as an active subject in their inbox, when that is not the case anymore. This may become a critical problem to solve as an organization grows and so the number of solicitations to its employees. The messaging pattern pursuits to finish that problem by imposing automatic rules that keep the talks flowing through the expected stages, ending as an archived subject. The goal is that users will define talk's course throughout an iterative path, recognizing it as an activity that has beginning, middle and an end.

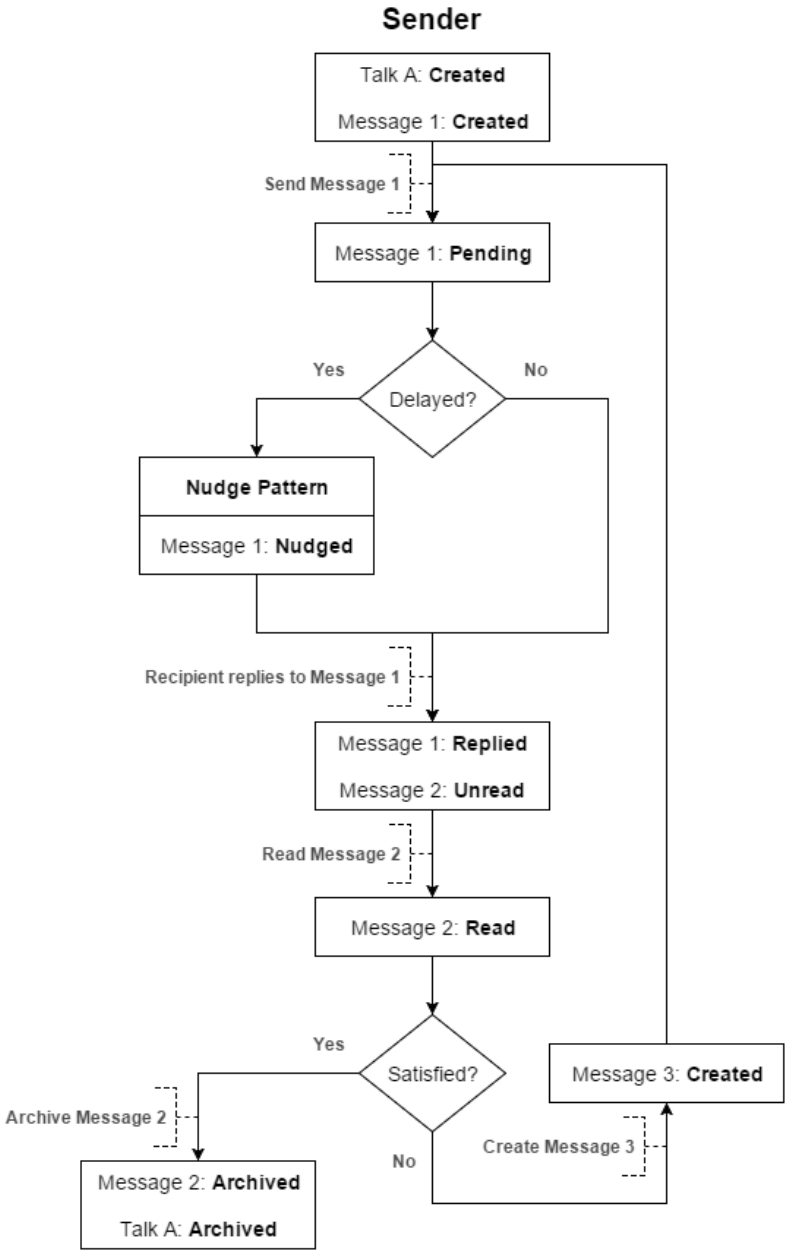

Figure 4.3: Messaging Pattern from Sender's View

The displayed sequence in the diagram starts with the creation of a talk, containing a new

message that is sent to one or more recipients transiting to a new state designated as pending. Taking into consideration the obligation of choosing a limit date to wait for a reply, the user may adopt one of two possible behaviors: either are nudging the recipient(s) as an alert for delay, or keep waiting for a reply in situations that are still on time. After that, and assuming the recipient replies to the first message, it is expected that the initial sender reads it and determines if the answer is suitable or not. If that is the case, then he is ready to archive not only the message but also the talk itself. More details about this functionality are provided ahead in section 4.3 regarding the archiving mechanism. In case the user is not satisfied with the received answer, he is able to continue the conversation with a new message, restarting the sequence.

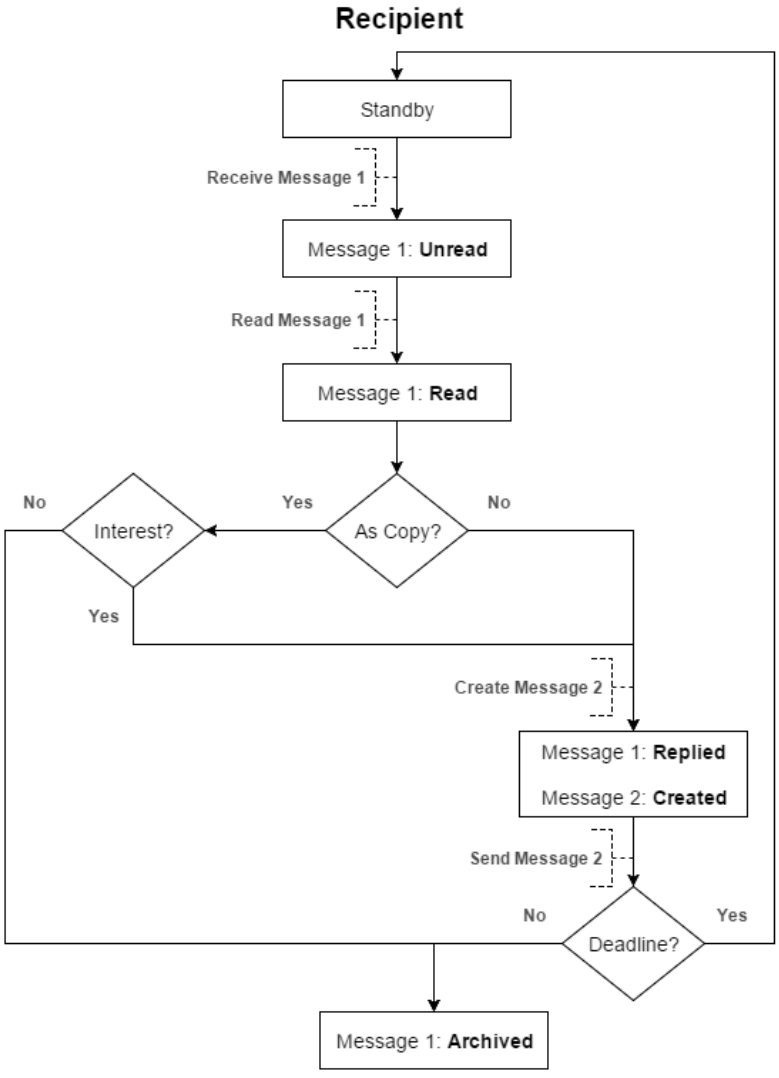

Figure 4.4: Messaging Pattern from Recipient's View

The recipient follows a straightforward and natural succession of steps, covering reception, reading and later replying the received message. However, the recipient may be receiving the message as a copy, equivalent to Cc or Bcc from e-mail, and here the possibilities are a consider-

ably different. Naturally, when as a main recipient (not in copy), the recipient must reply to the sender. The user receiving message as a copy has not that responsibility and may interfere only if he finds his input to be helpful or necessary. If no interest is shown by a secondary recipient (as copy), then the message received is automatically archived after reading.

Returning to the sequence, the recipient must pick a date as deadline, assuming he has some interest in keeping the conversation active. When that is not the case, the person may not select any date, considering that the answer provided concludes the conversation. In this case, the previous received messages are automatically archived. This action is also fully detailed in the next section.

- **Nudge Pattern**

In the course of a conversation there is a standard functionality that may produce a stimulus to keep the talk activity moving forward. The nudge pattern was projected with that purpose, and should be used as a surgical measure, avoiding its excessive use, since this may become ineffective in the long term.

The expected result, from this kind of interaction, is as simple and basic as highlighting, in some way, a message that still does not have reply after its deadline is exceeded.

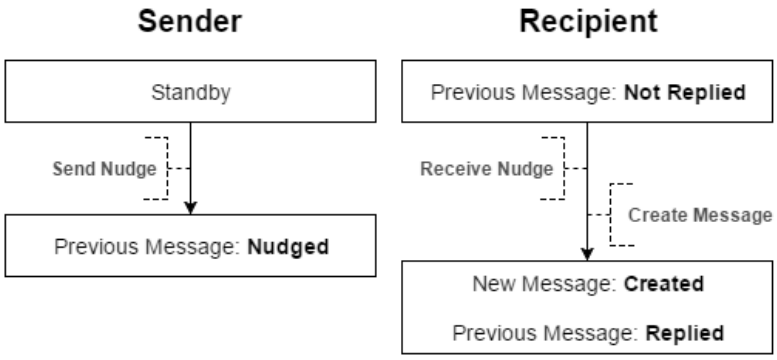

Figure 4.5: Nudge Pattern

This interaction is presented in Figure 4.5, considering a sender, the one who nudges, and a recipient that is nudged in order to get the necessary push to answer the received message.

- **Assignment Pattern**

As presented in Figure 4.6, the WIs have an associated implicit state to the assignment of a sender, for example an administrator or responsible for a department. This action requires an adequate and specific channel for communication purposes, as it will follow a pattern of interaction between the administrator and the member assigned to manage the WI. In this kind of scenario, it is evident that there are only two adequate options of final answer.

Accepting or declining an assignment is indeed one of the most critical situations inside an organization, since it represents the transference of responsibility. Being so, the system should

be prepared to facilitate this assignment management, imputing new responsibilities to the user assigned.

The assignment is associated to a specific WI, and the assigned state or not refers to that WI. Therefore, "Assignment 1" will be considered as a WI for a better schematics interpretation. As shown in Figure 4.6, the WI starts unassigned. This state passes to pending when a new person is selected to assume the responsibility of accomplish the respective WI. Once pending, it is expected an answer from the recipient, accepting, declining or postponing the decision. Postponing maintains the state as pending, while a definite answer pushes the interaction toward its end. those three options already mentioned will certainly lead to distinct situations and consequent actions.

Deciding whether accept or reject the assignment requires an insight on the pending situations that are still on recipient's hands. This person has always the possibility of postponing the choice, using the messaging pattern to explain the decision, but a final one must come up, at some point.

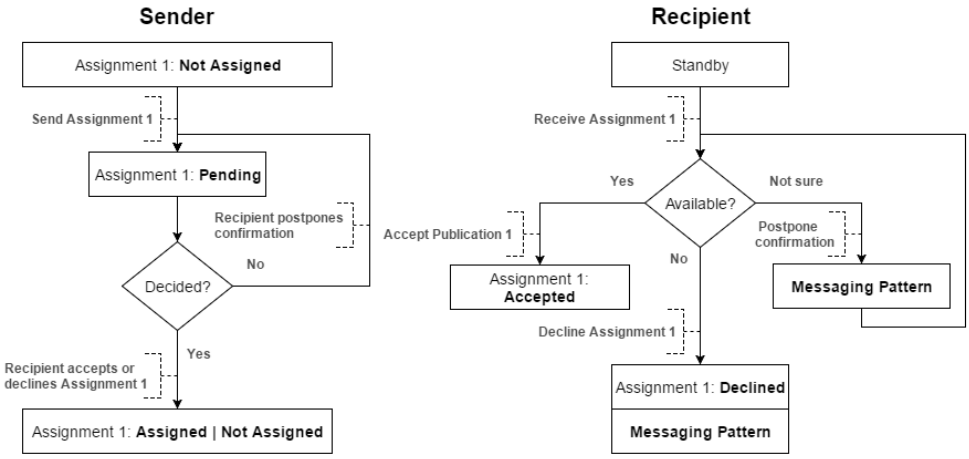

Figure 4.6: Assignment Pattern

When rejected, an explanation must be given to the sender, initializing again the messaging pattern. If accepted, no more actions are required or expected.

- **Invitation Pattern**

The objective of standardizing this pattern is essentially related to the massive message exchanging that will be generated from an invitation using a traditional method, expecting standalone replies from each recipient. It is intended to avoid that excessive amount of non-valuable information resulting from the multiple answers, taking advantage of simple responses as accept, decline or postpone.

As the previous pattern, the diagram presented in Figure 4.7, follows a similar sequence structure. Following the same idea, an invitation with one or multiple targeted persons is created at first. Then, the state of each invitation stays as pending. At this point, the sender will start having confirmations on people's availability to join the event. On the other side, the recipient receives the invitation and provides a definite answer or delays the decision, following the same principles presented before in other equivalent patterns.

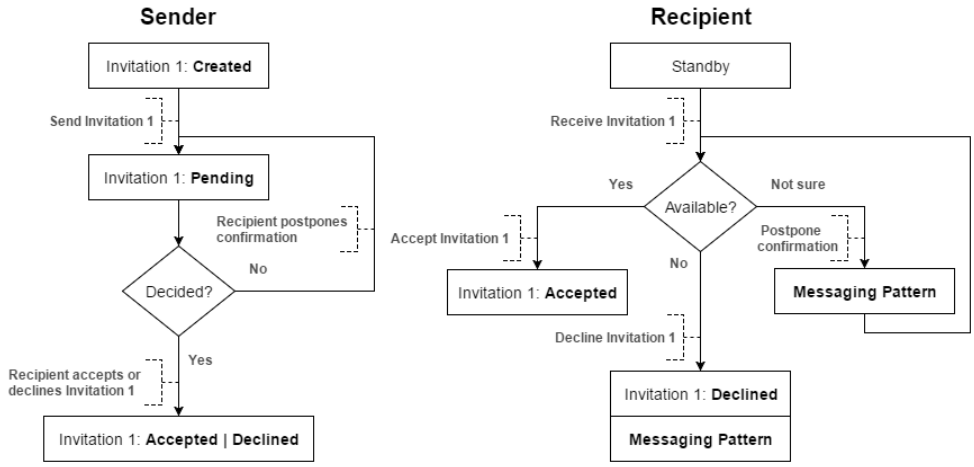

Figure 4.7: Invitation Pattern

The action of accepting or rejecting an invitation automatically affects the event's associated data, not requesting any further effort by the initial sender in updating the state of each invitation sent.

- **Validation Pattern**

Publication pattern comprehends the same logic as the assignment, since it has an equivalent set of actions and results. However, the context is slightly different and, for that reason, is understood as a typified interaction between users.

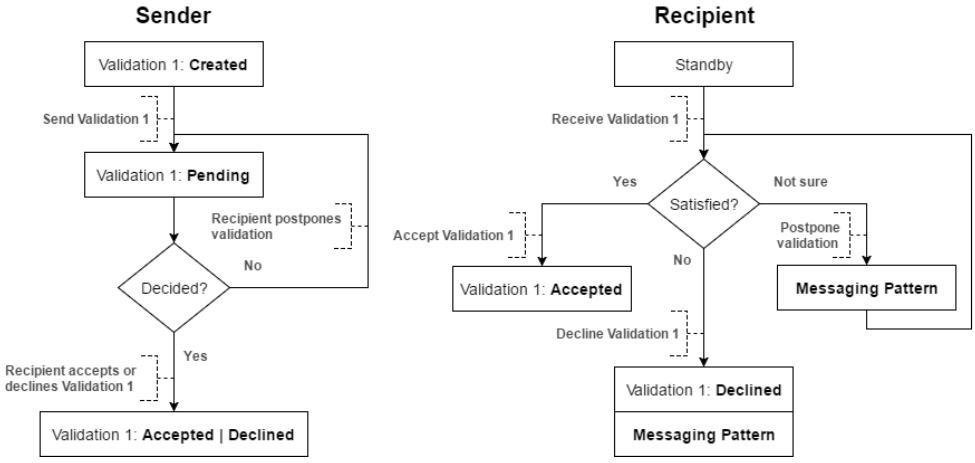

Figure 4.8: Validation Pattern

Initially, in Figure 4.8, a person creates a publication where a specific document is chosen as the target to be evaluated by the recipient(s). When sent, the publication's state is changed to pending, waiting for a decision on the validation of the proposed document. On the other side, the recipient must reach a conclusion and, depending on that, provide a justification for rejecting or postponing scenarios.

- **Notification Pattern**

This type of interaction often requires an acknowledgment of the notification reception. This is something that is intended to be an implicit action when reading the notification. The pattern here presented covers an internal perspective of the system. In posterior integration with e-mail, the referred acknowledgment is assured by a reading report.

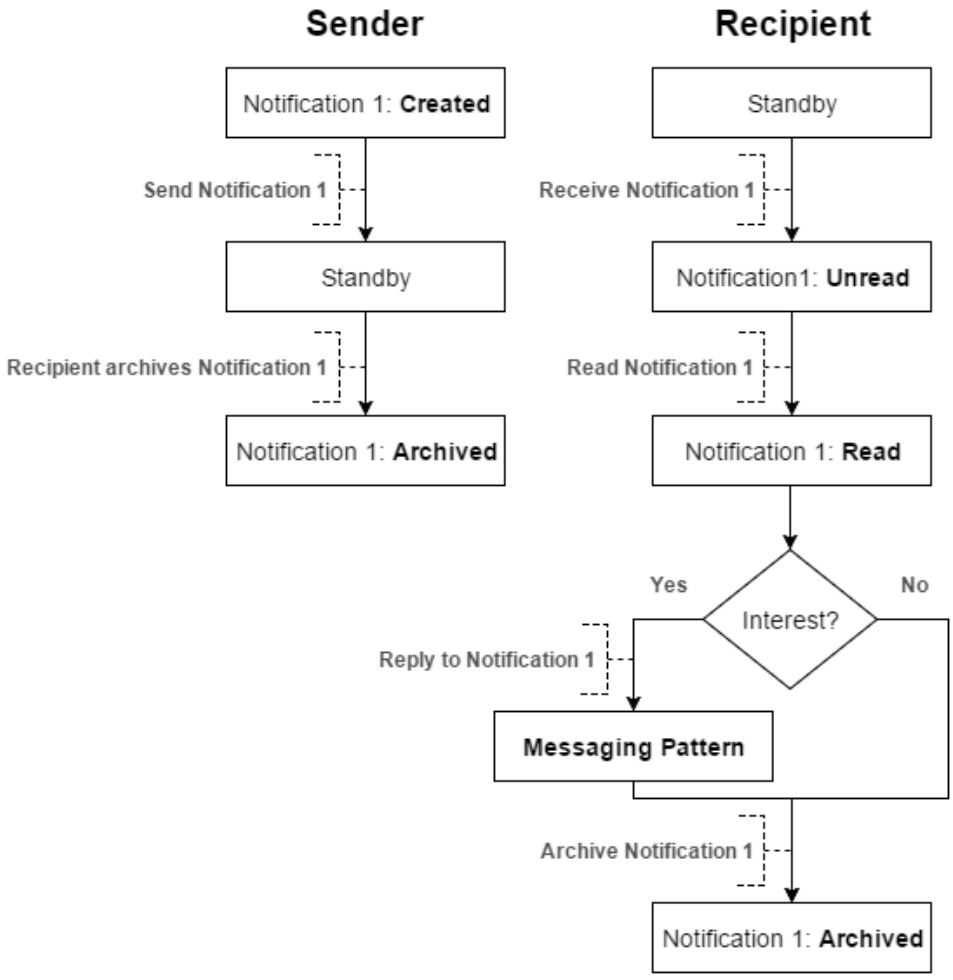

Figure 4.9: Notification Pattern

Observing the diagram above shown in Figure 4.9, two related paths are again described, regarding the sender's and recipient's view. The first one is a three-state diagram where a notification is created and sent before remaining in standby waiting for further advances. After this, and depending on the recipients' activity on this subject, the notification reaches its final stage, being archived.

On the recipient's side, the notification follows a natural sequence until it is read. Then, the recipient has a decision to make, pondering if he has interest in generating a new talk based on the received notification. This action contemplates the messaging pattern practice, previously

explained. Whether the user initiates a talk based on the notification or not it is expected that the notification culminates as archived.

- **Reporting Pattern**

The relations existent in different POOs may not be exactly the same, but the associated behavior remains intact, and for that reason, a pattern is proposed for this situation, reporting to a superior hierarchical level.

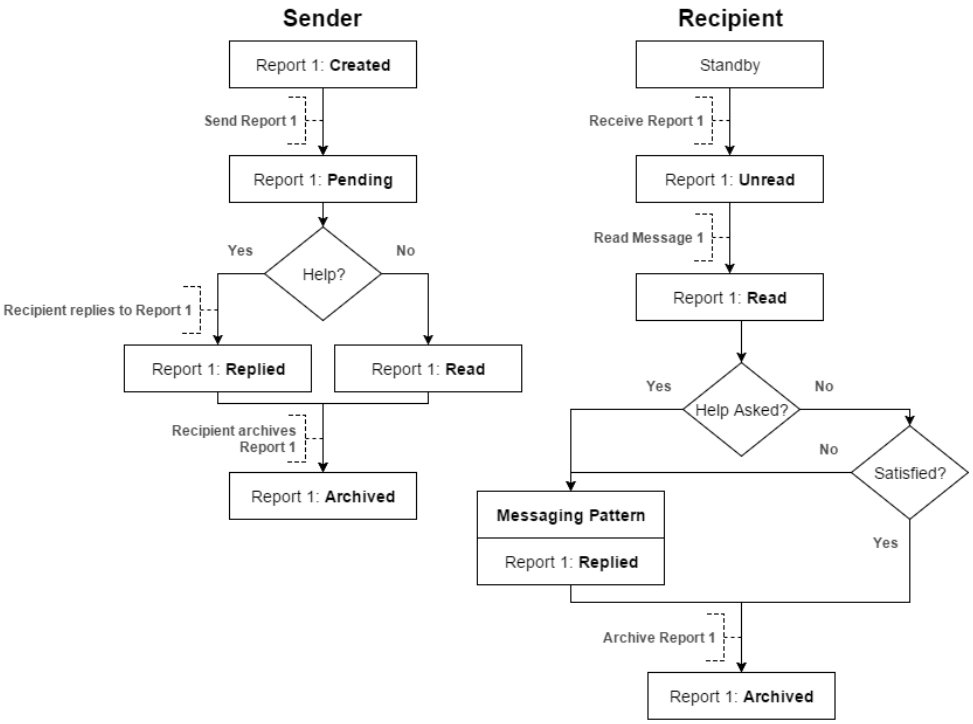

Figure 4.10: Reporting Pattern

As represented in Figure 4.10, the person reporting first provides information about the state of the projects, tasks and others on his hand, with the possibility of checking an option that informs the recipient that some kind of help is needed before finally submitting the report. If help was not requested, the report will, probably, be read and then archived. Now, considering help was demanded, the sender will be expecting that recipient reads his report and provides the necessary help by initializing a new talk with the adequate people for this purpose. A new conversation may be started whenever the recipient finds important his contribution in some of the matter reported. Dissatisfaction regarding the content of the report will be more likely the main motive to start a talk, even when not asked by the sender.

4.3 Communication Organizer

To support each pattern, presented in the previous section, several mechanisms and functionalities are developed. Beyond the messaging system itself, there are also a few concepts from GTD and PBD that will complement and make the system even more capable and powerful, boosting the productivity and objectives achievement.

Undoubtedly, the GTD methodology approaches topics with major importance on this matter, and the final solution built to manage the inbox includes those same principles intrinsically. Besides, the idea of one page shows all is applied in the inbox page, since it stands as a personal organizer that must facilitate the user's platform usage.

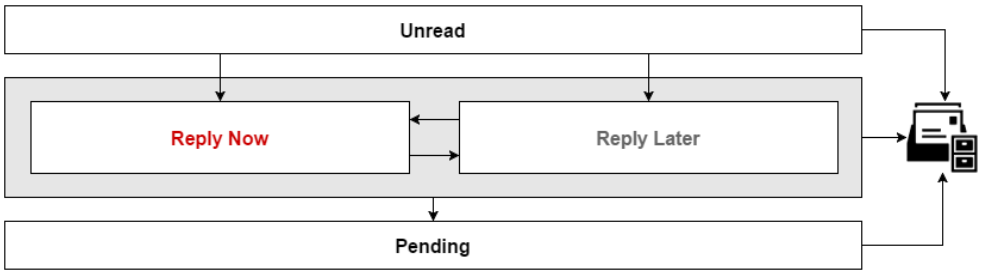

Figure 4.11: Schema of inbox page

As represented in Figure 4.11, there are four main containers allowing the management of the active talks. At the upper section of the page, there is an area identified as **Unread**, where new arriving messages are shown. The following functionalities are proposed to be applied in this section:

— open the complete talk in a popup in a single click;

— read the received message without having to open it;

— open a popup for a quick reply;

— mark messages as read in a single click;

— archive messages in a single click.

At the central section, one core block representing **Running** items may be divided in two separate areas identified as **Later** and **Now**, where messages already read will be displayed. The classification process for the messages depends only on the priority given to a specific subject by the recipient user. The following functionalities are implemented:

— drag and drop messages from one container to another;

— access to the WI associated, if that is the case, in one single click;

— all the functionalities mentioned in Unread's section, except mark messages as read, since they are already in that state.

As GTD defends, the idea of empty inbox is highly stimulated by the proposed system in both areas, Unread and Running, as well as prioritization of running items, providing a clearer vision of what is one user hands.

At the lower section, titled as **Pending**, messages sent and expecting new ones in return are presented. The following functionalities are proposed:

— read the message sent without having to open it;

— nudge other user(s), when a message has not been replied within the time interval previously defined;

— archive in one single click.

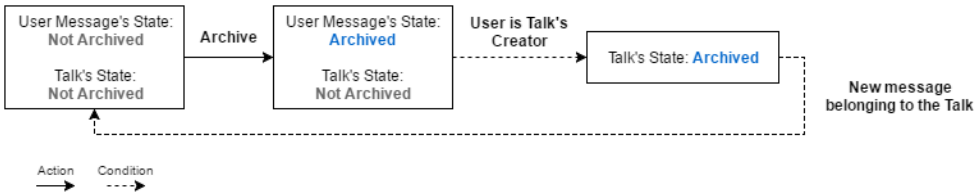

Figure 4.12: Archiving Mechanism Modelling

In Figure 4.11, the archiving mechanism is illustrated on the right side, representing a fundamental part of the proposed system. This mechanism is better described in Figure 4.12. As result of this instrument, the end-user will be capable of archiving talks with great agility in one single click.

Nevertheless, even providing simple tools that facilitate archiving, there will be always several users ignoring this functionality and keeping all the conversations in their inbox page. Thereafter, these people tend to represent a bottleneck in the processes where they take part, since their lack of organization may lead to low productivity and delays on responses to solicitations. The proposed mechanism prevents the possibility of this kind of situations happening with simplicity, usability and special attention to design aspects. The button for archiving is always displayed on the right side of the message, allowing the recipient to read the content, and then archiving it immediately, if that message closes the subject. Furthermore, the system has a built-in mechanism for automatic archiving when a user send a message without limit date to expect a reply.

This functionality is associated to each talk, and the result of executing this action is not only archiving the selected message, but also all other attached messages to the corresponding talk. Another scenario is also considered when a user, creator of a specific talk, archives his messages within this talk. In this cases, besides archiving his own messages, the user will be also archiving the talk itself, which means that conversation as reached its end and the obtained answers suit the initial problem or the user is not expecting any more replies. This leads to another topic, the continuity of a talk. For a better understanding of the concepts now explored, the properties of a talk are once more presented in Figure 4.13.

The conversation has three different stages. The first corresponds to the trigger that starts the talk itself, like a question or an information. After that, it is expected some sort of answer that fits the initial trigger. The final stage is the talk closure, where each user archives messages associated to the mentioned talk, assuming that no more relevant information is required or expected from talk's participants. Nevertheless, a talk conclusion does not represent inevitably that it cannot be

reopened again. If someone intends to add new information, the talk state will be changed to non-archived, and all the participants will receive related notifications again.

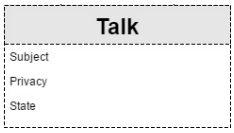

Figure 4.13: Talk properties

Integrating talks with other entities present in the system, like projects or users, opens an enormous range of relationships that allow to extract and track more complete and accurate information from the data source.

However, the permissions associated to each user will limit their access to data records regarding talks. As a standard procedure, talks are intended to be public whenever possible, what does not mean they have necessarily to be public. If the end user chooses private permission, then he will be limiting the ability from others, outside that talk, to view the contents shared. This possibility is probably more likely to be used in superior levels in the hierarchical structure of the organization.

4.4 Use Cases Matrix

A talk is constituted by three types of users which are a creator, an active and a passive member. The first one, is the one who triggered the conversation by sending the initial message. Active members are the ones who were added to the conversation as main (“To”) or secondary (“Cc”) recipients at some point of the talk’s timeline. Passive members are members that have some interest in a specific subject and started to follow the talk for that reason. These should never participate in a talk, unless they have special permissions for doing so.

To represent the main use cases, a matrix was prepared for each type of member, regarding the talk management, containing in green and blank colors the scenarios that are possible or not, respectively. The matrix is below represented in Table 4.1.

- **Use Case 1** — The user is able to send a new message inside the talk;
- **Use Case 2** — The user can archive all the talk received messages until that moment;
- **Use Case 3** — The user has permissions to archive the whole talk. This does not limit the possibility of sending new messages again, reopening the conversation;
- **Use Case 4** — The user is capable of receiving the messages associated to a talk where he is involved;
- **Use Case 5** — The user is able to read and access all messages within a talk;

- **Use Case 6** — The user may alert a recipient that he is late regarding a previous sent message.

Table 4.1: Talk Management Use Cases

Talk Management				
	Member			**Non-Member**
Public Talk	**Creator**	**Active**	**Passive**	
1. Send new message	■	■		
2. Archive previous messages	■	■	■	
3. Archive talk	■			
4. Receive messages in inbox	■	■	■	
5. Read messages	■	■	■	■
6. Nudge a member	■	■		
Private Talk				
1. Send new message	■	■		
2. Archive previous messages	■	■	■	
3. Archive talk	■			
4. Receive messages in inbox	■	■	■	
5. Read messages	■	■	■	
6. Nudge a member	■	■		

Chapter 5

Communication Organizer Implementation

This chapter contemplates the information regarding the implementation process. Initially, a brief introduction to the tools and libraries used in the implementation process and the application architecture is presented. After that, the database architecture is exhibited and explained, followed by the presentation of the developed webpages based on the communication organizer previously proposed.

5.1 Development Technology and Tools

During this Dissertation project, a set of tools were used to accomplish the established objectives. The technology selection, as foreseeable, was conditioned by the context were the final module was applied. Thus, the proposed solution was implemented using the same technologies employed by Dreamo in uONE platform development.

The software for development chosen was Microsoft Visual Studio Enterprise 2015, as it provides the necessary Integrated Development Environment (IDE) to develop a WBA like the proposed one, and Microsoft SQL Server 2014, as the database server necessary to store and provide information associated to the whole system.

On the subject of programming tools used, several were necessary for development purposes, and therefore are presented next. Primarily, it is important to clarify that the system implemented is a WBA, and both .NET framework and SharePoint platform from Microsoft were used to support developments. Furthermore, a set of programming tools were used and presented below.

A library offered by Telerik [29], designated as Kendo UI, was largely employed, since it provides a collection of controls, like grids, multiselect fields and many others, reducing the time needed for development. These controls will be presented throughout the next sections, as well as the final interfaces. The jQuery [30] library was mostly used to make Ajax calls, manipulate elements and for event handling.

Table 5.1: Programming Tools used in implementation

Stored Procedure	Using SQL, it allows to create standard queries that can be recurrently used by business logic.
C# LINQ	Both are used in the creation of methods in webservices that establish a connection with the database.
JS	Used to create dynamic webpages and establish the business logic of the communication module.
HTML CSS	Both are applied to generate the elements regarding the presentation layer.

5.2 Application Architecture

The implemented application separates the presentation layer from the business logic. All the data is on the server's side, which facilitates the database management, as well as the data access when required.

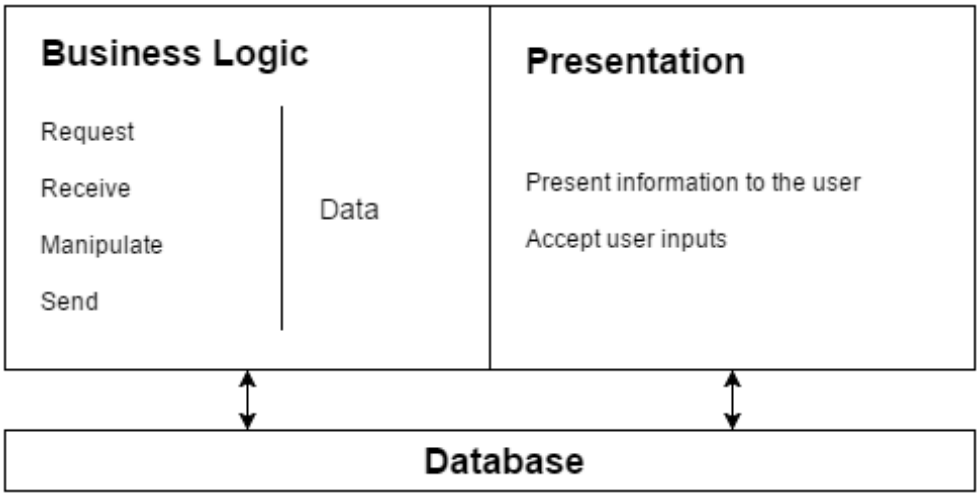

Figure 5.1: Application architecture schema

This kind of architecture, represented in Figure 5.1, provides a structure that supports parametrization of the developed web application, managing data in the source. This allow future updates in both layers without being restrained by existent data, that is exclusive for each client.

5.3 Database Architecture

Based on the conceptual model and the requirements established in chapter 3, a database structure is needed to support the projected system. Evidently, uONE had already several established entities and, for that reason, there was no need of creating all the entities used in communication module.

To better understand what was indeed developed for this dissertation purpose, only new entities created will be presented and fully detailed ahead, in section 5.3.1 and section 5.3.2.

5.3.1 Relational Model

Concerning the database structure and dependencies, it is unquestionable that all the entities are in some way related to the main element which is the Talk table. As a core part of this structure, the talk entity assumes a major role aggrouping the messages inside a conversation to only one key item. This aggregation brings multiple advantages to the system, allowing the management of conversations as one single piece and establishing relations that indeed increase the traceability of information for further searches.

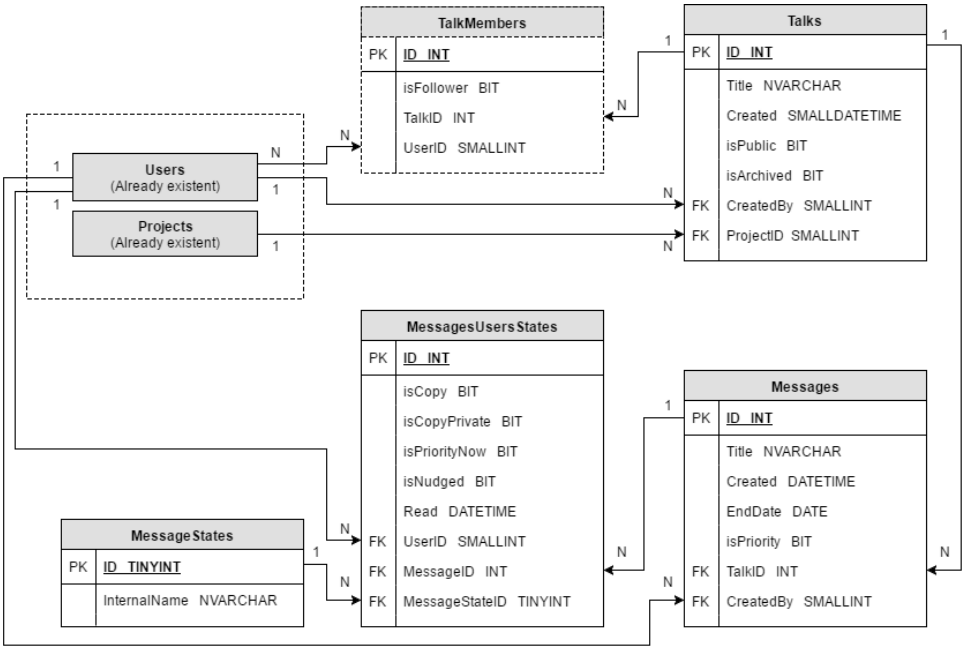

Figure 5.2: Relational Model

As exhibited above, in Figure 5.2, several tables were created to support the communication module. Both Users and Projects tables are displayed differently from others once they were already implemented before this project has started. However, since they establish associations with the remaining entities, its representation was considered important to a clear overview of the developed system.

5.3.2 Communication Organizer Entities

Every entity is used for supporting the communication mechanisms in the WBA. Nevertheless, each one has its purpose and relevance, affecting distinct pages and functionalities developed. For a better understanding of their contribution, a description of each one is provided next.

- **Talks**

As mentioned before, a Talk plays a major role in communication module for different reasons. First, since it aggroups the messages of one conversation like a master entity, the subject also stays linked to the talk instead of creating new ones to each message. The privacy associated to a conversation is defined here, as well as the state of the talk. One WI (Project) may also be associated to the conversation establishing an association to Projects' table.

- **TalkMembers**

This entity allows the creation of groups of users that relate to a specific talk. A special attribute, regarding the type of user associated, is here presented as "isFollower". It determines the difference between active and passive members inside a talk.

- **Messages**

Represents the items contained by each talk, since no message can be created aside from it. The priority here represented is the one selected by the sender when sending a new message. Also, the limit date ("EndDate") expected for replying to one message is contemplated in this entity.

- **MessagesUsersStates**

For each recipient of a specific message, several information must be known to assure a proper behavior of the system. Thus, this is one of the most critical entities, once it storages the largest quantity of data, in this module. Here is characterized the expected role of the user by identifying the message as copy or not, as well as the message's state for him.

- **MessageStates**

This entity was created with the purpose of allowing a dynamic parametrization of the data regarding the state of a received message, when applicable.

5.4 Inbox

The Inbox area is considered the core implemented webpage due to the multiple functionalities and features offered to the end-user. As mentioned previously, in chapter 4, it was intended to be a sort of personal organizer, displayed in Figure 5.3, that allows users to manage their incoming messages and talks, where the usability and simplicity of this page is an essential requirement.

To achieve the goals defined, Kendo controls were used to develop the main functionalities present here. Thus, KendoGrids are used to list messages in four containers, regarding "Unread", "Later", "Now" and "Pending" items. Starting by the "Unread" section, it is possible to notice that the layout was minimized as much as possible, keeping only the necessary information only. This principle was used in all the created grids as it decreases the amount of non-value added

contents or relevant information to the user. Hence, the messages incoming are instantly grouped by project (WI), concentrating all these messages, even in separate talks, which facilitates the perception of what is taking place on each project. The management of the multiple conversations is also improved and way easier to do.

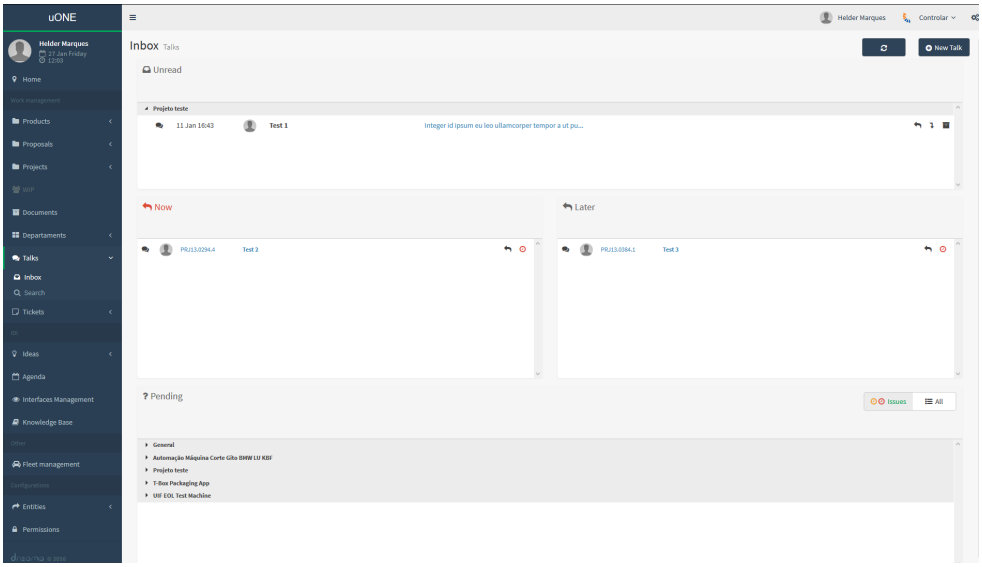

Figure 5.3: Personal Organizer

Each row concerns to a received message, providing some details such as date and time of reception, the sender of the message, its subject, a preview of the message and some buttons that allow to execute four distinct actions:

- **Open Talk**: open a popup which presents more details associated to the talk along with the messages formerly exchanged. This feature is presented in section 5.4.1.
- **Open Quick Reply**: open a popup with a form to send an instant reply. This feature is presented in section 5.4.2.
- **Mark as Read and Update Personal Priority**: affect immediately the data in the database by updating the state and the priority of the received message.
- **Archive**: mark the message and all the previous ones as archived. If the person who archives is the one who started the talk in first place, then the archive action will modify the state of the talk to archived as well.

The central area exhibits messages marked as read are shown. Here, the organizer takes a major role since it allows the drag and drop of items from one side to another, changing the priority of the message targeted to “Now” or “Later”.

To implement this functionality, both Kendo Draggable and DropTarget controls from were used simultaneously with Ajax calls to update data instantaneously in data source. One example of this kind of implementation may be seen in Figure 5.4.

```
$("#_gridTalksRunningNow").data("kendoGrid").table.kendoDraggable({
    filter: "tbody > tr",
    group: "gridGroup1",
    hint: function (e)...,
});
$("#_gridTalksRunningNow").data("kendoGrid").wrapper.kendoDropTarget({ //Move from "Later" to "Now"
    drop: function (e) {
        var _target = $("#_gridTalksRunningLater").data("kendoGrid").dataSource.getByUid($(e.draggable.currentTarget).data("uid"));
        var _messageID = _target["ID"];
        SetMessagePriorityNow(_messageID, 'toNow')
        $("#_gridTalksRunningLater").data("kendoGrid").dataSource.read();
        $("#_gridTalksRunningNow").data("kendoGrid").dataSource.read();
    },
    group: "gridGroup2"
});
```

Figure 5.4: Controls used to implement the drag and drop functionality

Besides allowing quick reply and opening the talk, this container also provide information to the user of how many days he has or already is past due from the deadline to answer that specific message. Two days before it reaches the deadline defined by the sender, the color changes to yellow, and ends up changing to red when it reaches it.

Finally, at the bottom of the page a new container, includes the sent messages expecting a reply, is displayed. Again, grouping is used to facilitate the interpretation of the items show immediately bellow. As the remaining grids, the functionalities present in each row are similar. However, quick reply is not presented, as it probably would not be used in these cases. It is important to refer that the "Check" symbol at the end of each message preview indicates if the message is read or not by the recipient.

Toggle buttons are used here to separate messages exceed the limit date defined, or that may reach it soon, from all messages sent and still wait for answer.

Furthermore, the page is refresh each minute to keep user updated without having to force refreshing. However, if that is the case, the user may force the refresh in the four sections by clicking on the left button on top right.

5.4.1 Talk

To send a new message, and consequently create a new talk, the user enters the page below shown in Figure 5.5. This is a simple form where typical data is required to send the message.

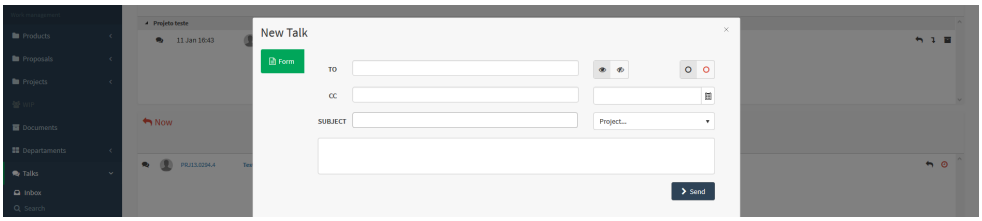

Figure 5.5: Form that starts a new talk

When the user intends to read all the exchanged messages in a conversation it is that he access the talk form page, shown in Figure 5.6. For that purpose, a new grid was created to display all the messages, showing firstly the most recent added messages.

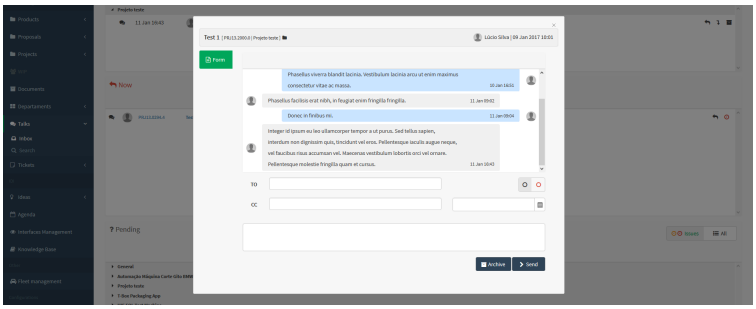

Figure 5.6: Presentation of messages associated to a talk inside a popup

On the popup's top area, a few details about the talk are displayed, such as the associated WI, the talk's subject, the person who created, and when it was started. In cases where no WI is linked to the conversation, a dropdown list with all the projects available to select is shown. The database will be updated, as soon as one project is chosen.

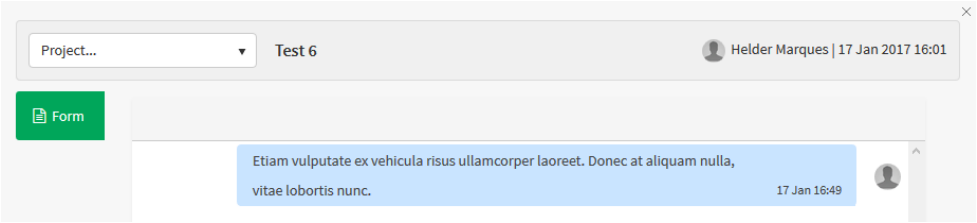

Figure 5.7: Dropdown button to select a project and the remaining form details

To elaborate the form shown in Figure 5.7, other controls from Kendo UI were used, such as Multiselect, for selecting the main and secondary recipients, Button to choose the talk's privacy and priority, and the Datepicker that allows to decide the deadline.

5.4.2 Quick Reply

This form only allows the insertion of a new message in database, assuring a quite simple and immediate tool to facilitate the communication process. Analyzing Figure 5.8, this become more evident, as only message input, deadline, if necessary, and priority are available to the user.

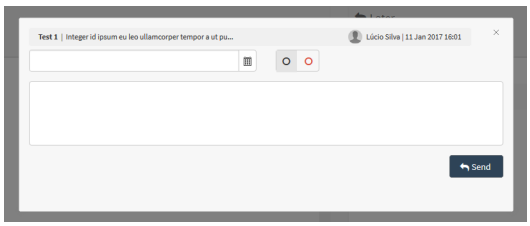

Figure 5.8: Form for quick reply

In most scenarios, it is projected that this kind of solution will be vastly used by end-users, as it will provide a quick method for sending new messages inside a talk already ongoing.

5.4.3 Nudge

Based on the ContextMenu control from Kendo UI, the action of nudging other users, shown in Figure 5.9, was developed providing a new mechanism for alerting the recipient that the sender is still waiting for a reply on a specific subject. This control instantaneously updates the state of the message received as nudged, showing to both users a sign of that current state.

Figure 5.9: Nudge action

5.5 Search

Regarding traceability of information, two new main pages were developed to simplify the search of archived and active talks. The first refers to talks in general, reaching all the conversation that the end-user has access. Then, a new page, based on the same principles as the first one, was implemented within the main page of project management. Both situations are presented in the next sections.

5.5.1 Talks Overview

As previously mentioned, the purpose of this page, displayed in Figure 5.10, is to concentrate the different conversations that the user has access.

However, some measures had to be taken to prevent situations that may lead to slowness due to the quantity of information transferred from the server to the browser. Thus, three main decisions were taken for data requesting:

- **Start empty**: when opening this page, no data is regarding messages is presented.
- **Show on demand**: messages are presented on demand, avoiding to overload the page with needless information.
- **Dual data source**: when picking the toggle button of “Personal” or “All” talks, the user will be selecting the data source for his search. In the first case, this conditions limits the data

transferred to only conversations where the user take part. Instead, the option “All” provides these messages plus the others where he does not take part but their content is public.

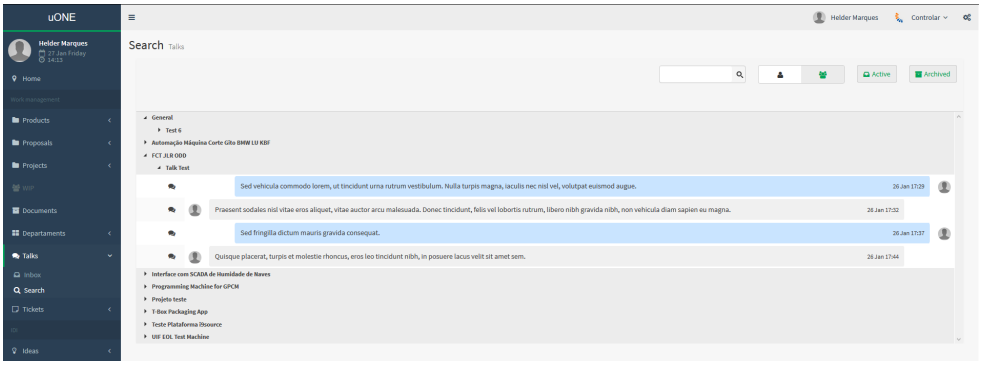

Figure 5.10: Presentation of Search page

This area is oriented to present talks and respective messages, not proving any possible action that may affect the database. Indeed, the arduous work to implement this page regards the configuration and elaboration of filters capable of allowing the accurate visualization of data for each option possible. The search field accepts inputs from the user, and presents the results as he writes, improving significantly the user experience.

For messages presentation, when selecting one or more available filters, all the messages are grouped by talk and project reducing the quantity of information displayed to the user. However, there is always the option of expanding each grouped row to view exchanged messages. Also, when searching for something, the rows are automatically expanded to facilitate the results analysis. The function that provides this functionality is demonstrated in Figure 5.11.

```
function ExpandCollapseTalksGrids(GridID: string, valueAction: any): void {
    try {
        if (valueAction == ExpandCollapseGroups.ExpandAllGroups) {
            $("#" + GridID).data("kendoGrid").table.find(".k-grouping-row").each(function () {
                $("#" + GridID).data("kendoGrid").expandGroup(this);
            });
        }
        if (valueAction == ExpandCollapseGroups.CollapseAllGroups) {
            $("#" + GridID).data("kendoGrid").table.find(".k-grouping-row").each(function () {
                $("#" + GridID).data("kendoGrid").collapseGroup(this);
            });
        }
    }
    catch (exception) { CoreException('uONE', 'ExpandCollapseTalksGrids', exception); }
}
```

Figure 5.11: Generic function that allows to expand or collapse the rows

5.5.2 Integration with Work Items

Each project has a considerable amount of information related to documents, tasks, and others. As planned, conversations also take an essential role in projects progression and the information

exchanged within a project, if identified as public, must be easily found. For that reason, a new tab was added to the project management page, presented in Figure 5.12. The latter has a similar layout, as well as functionalities. However, in this case, the database provides information limited to the concerned project.

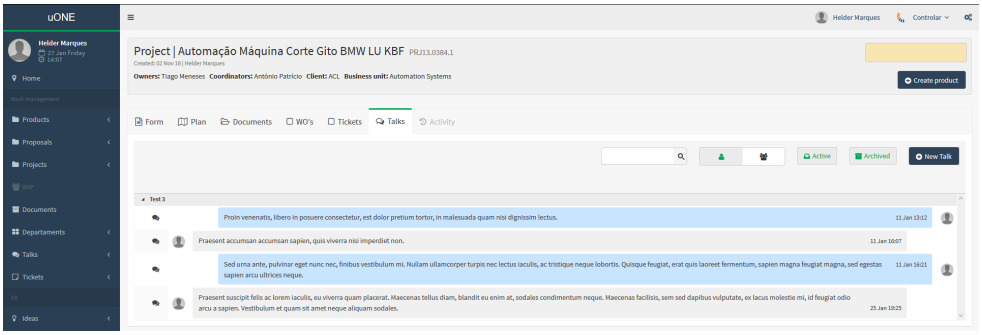

Figure 5.12: Presentation of messages associated to a talk inside a project's page

Chapter 6

System Testing and Validation

This chapter aims to describe the tests and validations used to guarantee the quality of the system accomplished during this master dissertation. Therefore, a demonstration of a talk life cycle is presented, followed by a brief description of how the system was integrated and tested by a team from Dreamo. After that, validation scenarios are presented.

6.1 Talk Flow Testing

During the project, several tests were done to validate the module settled inside the uONE platform, and for a better understanding of what indeed occurs from the creation until the archive of a talk, a sequence is exhibited below.

With this sequence example, it was intended to demonstrate a typical test done to verify that every talk follows the according course, presenting at each time the accurate data, regarding the distinct actors and their interactions.

Initially, the User A starts a talk designated as "Talk Test" by opening a new talk form and fulfilling the several fields with the necessary data to send the initial message, as shown in Figure 6.1.

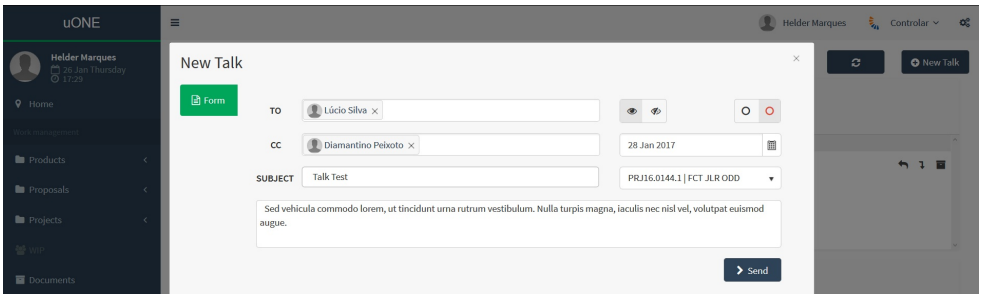

Figure 6.1: Starting a new talk named "Talk Test"

When the message is sent, new rows related to the messages sent to main recipients, of the recent created talk, are added to the Pending's section, as displayed in Figure 6.2.

As expected, the message sent to User C in Cc is not displayed in that section, as no reply is expected from him.

Figure 6.2: Message sent in the sender Pending's box

At this point, as exhibited in Figure 6.3, the User B receives the message sent by User A, being this included in his Unread's section.

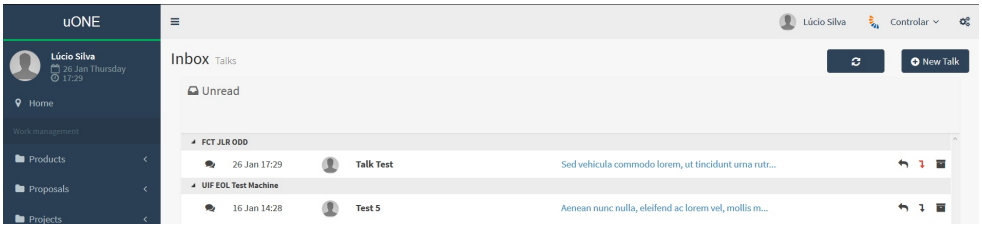

Figure 6.3: Message shown in the recipient's inbox page

After receiving the new message, the User B opens the popup where the conversation is fully detailed, as seen in Figure 6.4.

At this moment, the User A can verify that User B has read the sent message since the check symbol on the right side has changed from dark blue to green, as seen in the Figure 6.5.

After opening the talk form, User B replies to User A establishing a new limit date to get an answer.

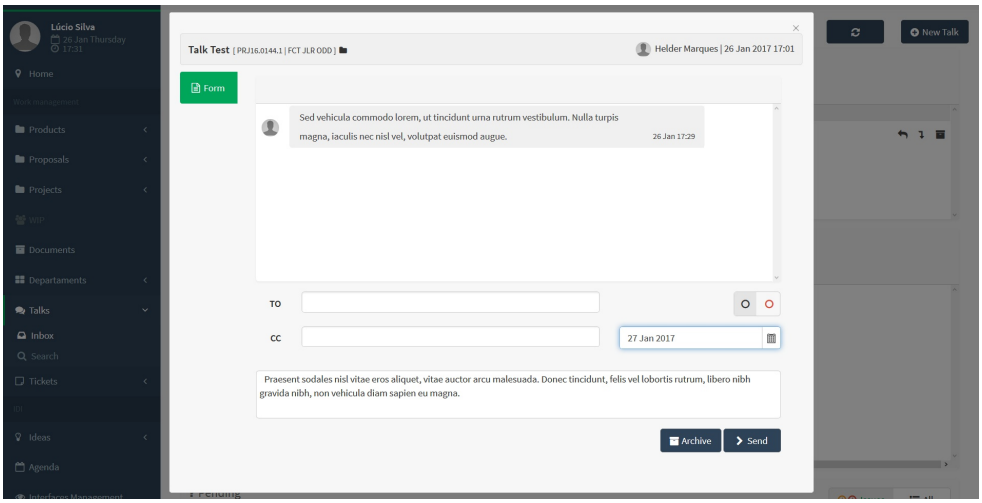

Figure 6.4: Creating a new message in the Talk form

Figure 6.5: Message sent updated in the sender Pending's section

Hence, the User A receives the message sent by User B and replies to it, using the quick reply form, as shown in Figure 6.6 and 6.7.

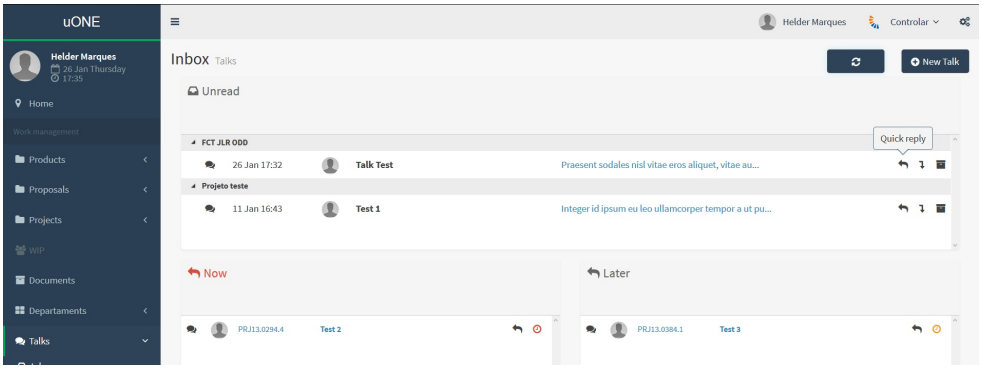

Figure 6.6: Creating a new message in the Quick Reply form

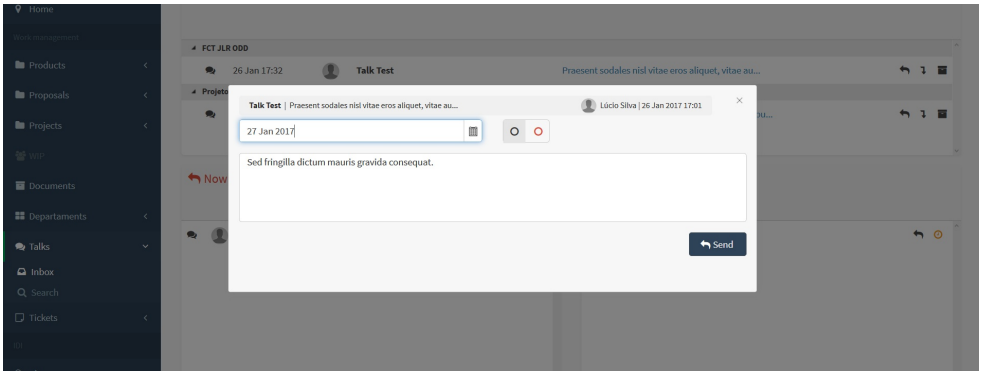

Figure 6.7: Sending an instantaneous message

The User B receives the message sent by User A and marks it as read, transiting from the Unread's container, in Figure 6.8, to the Later's one, in Figure 6.9.

Since the message has a normal priority, previously attributed by User A, the action of marking it as a read item forces the incoming message to be moved to the Later's section, since this area is reserved for situations where there is no need to act immediately.

This automatic behavior does not prevent, in any way, the possibility of updating later the personal priority of a specific message, if the recipient thinks it is the most appropriate action to

take.

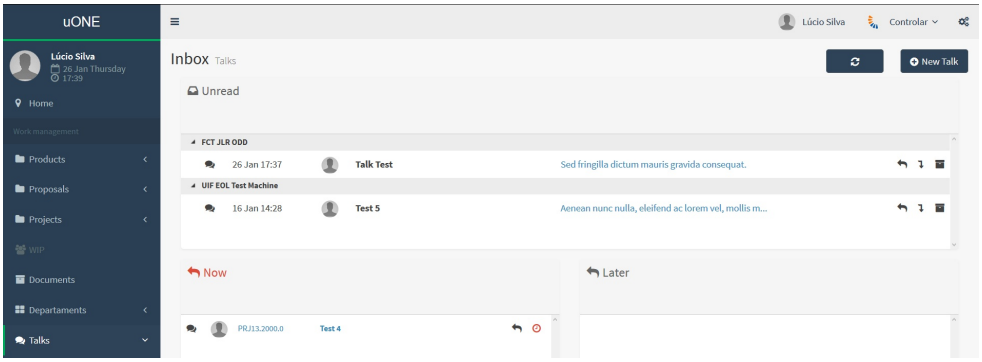

Figure 6.8: Receiving a new message

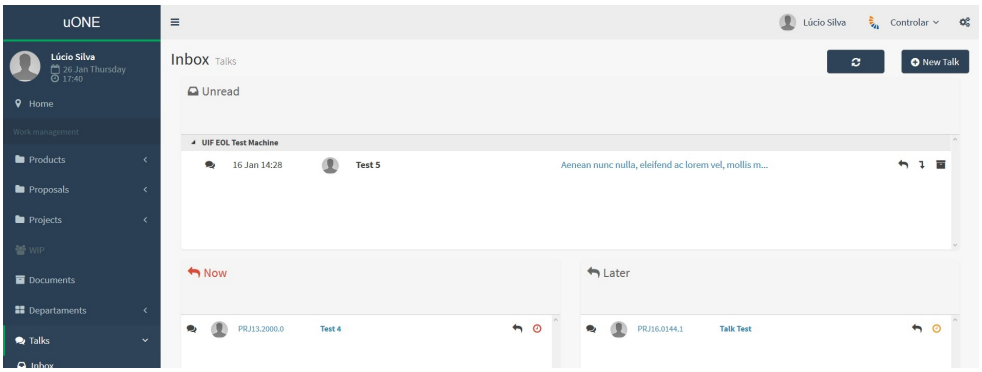

Figure 6.9: Message read in Later's box

However, in this particular case, the User B assesses the received message from User A as an important matter to resolve, and for that reason he drags it to the Now's section. This situation is represented in Figure 6.10.

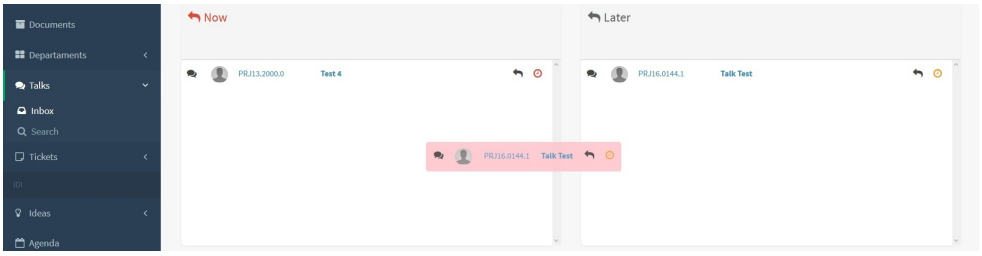

Figure 6.10: Drag and Drop between the two central containers

Following the usual pattern, the User B replies to User A by opening the talk form and filling the text area with the message's content, as presented in Figure 6.11.

At this point, User B does not define any specific date for reply, since he considers that his replay will suit the recipient's solicitation.

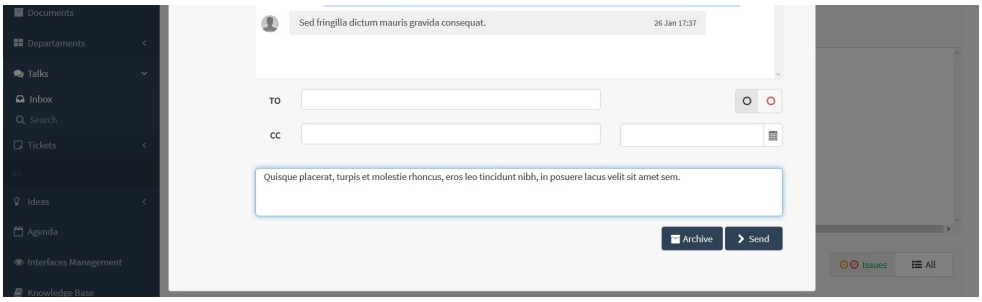

Figure 6.11: Replying to the message received

As Figure 6.12 aims to represent, User A considers the conversation as concluded with the last received response and have no more relevant information to add to the conversation. Thus, he archives it and the messages associated to him and the talk itself are updated to the state mentioned.

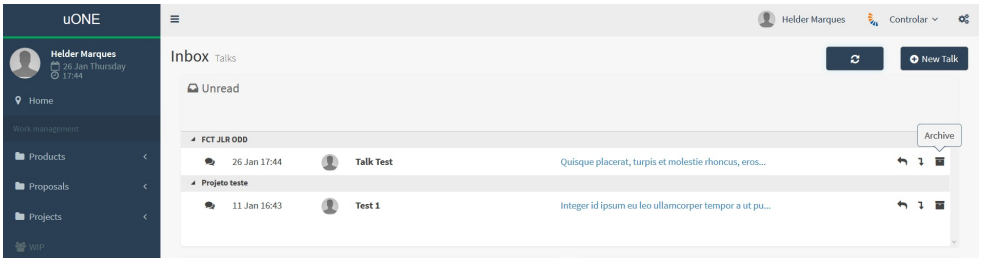

Figure 6.12: Archiving action triggered

If a user intends to find the conversation before presented, he may start inserting some inputs related with the talk, in the search field, exhibited in Figure 6.13.

As expected, the conversation will be fully shown if the data submitted does have some relation with information as the subject or the people involved.

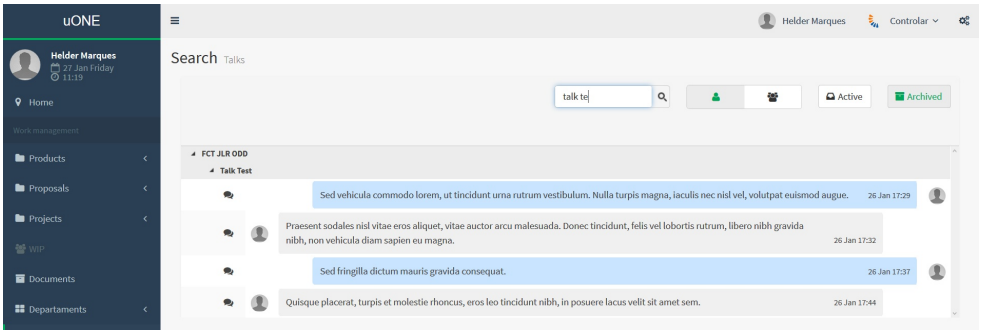

Figure 6.13: Searching for a conversation

It is expected that at any time, User A nudges another user for a message that is still waiting for a reply. In that case, User A opens the context menu, as shown in Figure 6.14, and clicks “Nudge”

item. Then, the recipient will notice that the sender's picture in the nudged message will have a thin red line surrounding it, as shown in Figure 6.15.

Figure 6.14: Nudging other user

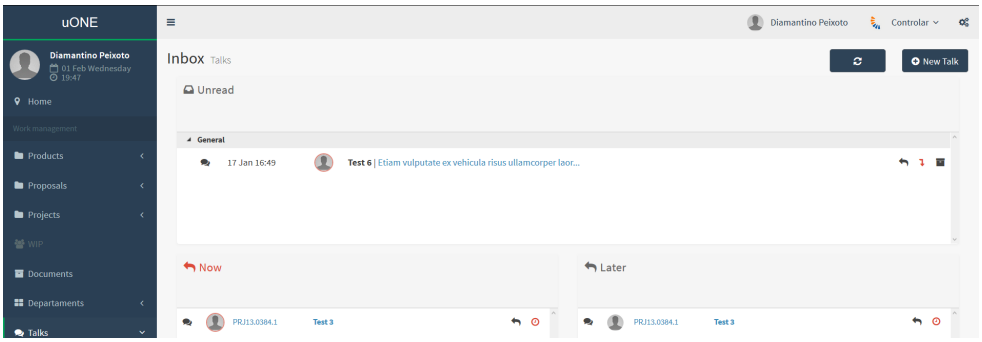

Figure 6.15: Presenting the nudged message

6.2 Technical Integration

Concerning the integration of the developed module in the final solution of uONE, several tests and verifications were performed by Dreamo developers.

This module was firstly built on a preliminary solution provided by Dreamo and, in order to integrate in the final platform, was necessary to test all the implemented functionalities and assure that all the source code was in harmony with the rest of the application. Besides, the database updates were verified to guarantee that all the created structures were accordant with the requirements defined by the team, and ready to be copied to the final database.

After this stage, the integration, regarding technical aspects, was considered successful, and no need of modifications was identified.

6.3 uONE Team Validation

After the final integration of the functionalities implemented in the preliminary solution, the uONE team's responsible had to validate that the created layouts followed the same patterns as the ones already used on the platform. This validation occurred without any significant change, since only

a few corrections in the main CSS file were done. Besides, the usability present in this system also suited the established requirements as well as the operability of the system in general, in what concerns the use cases.

Thus, the developed module was approved and validated as suitable to the end-user, and consequently to future clients.

6.4 Independent End-User Validation

The application was also tested by a group of independent persons, to guarantee that the usability and the simplicity imposed are adequate to its final purpose. These tests occurred after the integration in uONE platform has been completed, assuring that all the functionalities were fully operable.

By the end of this assessment, the system was validated by the user, only pointing some possible enhancements for posterior development.

Chapter 7

Conclusions and Future Work

Taking into consideration the topics previously discussed, several considerations regarding the overall work developed and results obtained must be now presented and analyzed.

There were two main objectives for this dissertation project, the analysis and identification of communication patterns that can be applied in project-oriented organizations and the development of a web-based communication module that supported the personal communication management of each organization's member.

For the first objective, an extensive solution was proposed, it took in consideration the common interactions involving people and it contemplated the seven key patterns described in chapter 4.

Regarding the second objective, a system capable of supporting the communication flows present in a project-oriented organization was developed, taking into account the requirements defined in chapter 3 and the solution proposed in chapter 4.

Based on the built solution, it is possible to identify three core ideas that certainly establish a significant difference when comparing with other solutions available in the market, presenting a new perspective for the communication process. The first difference is that every conversation should be created with a clear purpose, and once satisfied, its archiving is necessary. The second one refers to the personal organizer which provides to the end-user the ability to manage his own solicitations in a single page. The third and last important idea to refer regards the traceability of information, that must be accomplished using a complex database background solution, which does not confuse or intimidate the end-user due to excess of information required.

Developing the communication management application lead to numerous questions, mainly concerning the properties of each conversation as the main element of the whole module. Once established, all the mechanisms necessary for communication management, such as archiving and prioritizing conversations, became easier to accomplish.

Finally, the preliminary developed system was validated by Dreamo and integrated in the uONE platform, concluding the implementation part of this dissertation project.

During the system development, several obstacles arisen essentially due to some lack of experience in programming within a complex environment with multiple dependencies and restrictions

existent from the general context of the application. Nonetheless, all the difficulties were overtaken with success.

Examining the accomplishments obtained this MSc dissertation project it is possible to consider that the main objectives were indeed achieved, evidencing the validity of the proposed system.

7.1 Future Work

Several perspectives for future developments may be considered as essential to obtain a fully integrated and standardized system.

The communication patterns identified, as normalized interactions, must be prepared and transformed into entities to be integrated in the system afterwards. Integrating these would certainly increase the platform's value, providing to its users, tools that will simplify and make their communication channels more effective. Besides, the patterns also would have a considerable impact in the integration between conversations and the remaining entities present in the system. Examples of this situation might be the association of all types of work items to a conversation and the possibility of pointing one or more documents to send within a talk.

Increasing the usability is another key point to enhance the simplicity of the system, while increasing the functionalities that it provides to the end-user. About this topic, context menus could be used to group a set of possible actions related to the selected item, keeping the interface simple and clean, as the context menus only appear when the user intends to.

The permissions management is also a matter that still needs a few enhancements concerning the presentation of specific details, when receiving a new message, and possible actions, according to the users' permissions. Analyzing the communication patterns previously identified, it is possible to understand that each one of them is applied in distinct scenarios that would variate depending on the users' role.

An additional improvement in the archiving mechanism already developed must be performed, allowing the automatic update of a talk's state to archived when a fixed period of inactivity is reached.

Another suggested feature would be the integration with e-mail, once that it would facilitate inevitably the exchanging of messages with the external clients. This implementation is indeed considered as a key upgrade to be done in short-term, since it is highly required by Dreamo's clients.

As the developed system had the purpose of increasing productivity of each person, a possible situation that may be considered, as it has some interest, is to create a parallel system of key performance indicators that uses specific data from the entities involved in the communication module, providing real time information of their activity. Besides, an overview of the whole communication system operation would be provided to superior levels of management within an organization.

References

[1] EPM Live. Epm live | social collaboration. URL: http://epmlive.com/products/work-management/social-collaboration/ [last accessed 2016-07-16].

[2] Microsoft. Yammer enterprise social network | microsoft office 365. URL: https://products.office.com/en-US/yammer [last accessed 2016-07-18].

[3] Gareth R Jones. *Organizational Theory, Design, and Change: Text and Cases*, chapter 10. Pearson Education, 4th edition, 2003.

[4] Roberto Evaristo and Paul C van Fenema. A typology of project management: emergence and evolution of new forms. *International Journal of Project Management*, 17(5):275 – 281, 1999. URL: http://www.sciencedirect.com/science/article/pii/S0263786398000416, doi:http://dx.doi.org/10.1016/S0263-7863(98)00041-6.

[5] Maria Ester Ferreira and Anabela Pereira Tereso. *Software Tools for Project Management – Focus on Collaborative Management*, pages 73–84. Springer International Publishing, Cham, 2014. URL: http://dx.doi.org/10.1007/978-3-319-05948-8_8, doi:10.1007/978-3-319-05948-8_8.

[6] Kathy Schwalbe. *Information Technology Project Management*, chapter 10. Cengage Learning, 7 edition, 2013.

[7] B. Ramesh. Process knowledge management with traceability. *IEEE Software*, 19(3):50–52, May 2002. doi:10.1109/MS.2002.1003454.

[8] David Allen. *Getting Things Done. The Art of Stress-Free Productivity*. Penguin, 1 edition, 2003.

[9] Irja Hyväri. Project management effectiveness in project-oriented business organizations. *International Journal of Project Management*, 24(3):216 – 225, 2006. URL: http://www.sciencedirect.com/science/article/pii/S0263786305000955, doi:http://dx.doi.org/10.1016/j.ijproman.2005.09.001.

[10] Louis Raymond and François Bergeron. Project management information systems: An empirical study of their impact on project managers and project success. *International Journal of Project Management*, 26(2):213 – 220, 2008. URL: http://www.sciencedirect.com/science/article/pii/S0263786307000981, doi:http://dx.doi.org/10.1016/j.ijproman.2007.06.002.

[11] Project Management Institute. *A Guide to the Project Management Body of Knowledge: PMBOK*, chapter 1. Project Management Institute, 5 edition, 2013.

[12] Hans J. Thamhain. *Team Building in Project Management*, pages 823–846. John Wiley & Sons, Inc., 2008. URL: http://dx.doi.org/10.1002/9780470172353.ch32, doi:10.1002/9780470172353.ch32.

[13] Brian Helbrough. Computer assisted collaboration — the fourth dimension of project management? *International Journal of Project Management*, 13(5):329 – 333, 1995. URL: http://www.sciencedirect.com/science/article/pii/026378639500041N, doi:http://dx.doi.org/10.1016/0263-7863(95)00041-N.

[14] Kathy Schwalbe. *Information Technology Project Management*, chapter 1. Cengage Learning, 7 edition, 2013.

[15] Roland Gareis David Cleland. *Global Project Management Handbook: Planning, Organizing and Controlling International Projects, Second Edition: Planning, Organizing, and Controlling International Projects*, chapter 2. McGraw-Hill Professional, 2nd edition, 2006.

[16] Project Management Institute. *A Guide to the Project Management Body of Knowledge: PMBOK*, chapter 10. Project Management Institute, 5 edition, 2013.

[17] W. Qian and S. Zhen-hua. Research on multi-perspective communication management of software development project based on theory of project management. In *Signal Processing Systems (ICSPS), 2010 2nd International Conference on*, volume 3, pages V3–192–V3–195, July 2010. doi:10.1109/ICSPS.2010.5555844.

[18] Adedeji B. Badiru. *Triple C Model of Project Management: Communication, Cooperation, and Coordination (Industrial Innovation)*, chapter 2. CRC Press, 2008.

[19] Terry Winograd. A language/action perspective on the design of cooperative work. *Hum.-Comput. Interact.*, 3(1):3–30, March 1987. URL: http://dx.doi.org/10.1207/s15327051hci0301_2, doi:10.1207/s15327051hci0301_2.

[20] BJ Fogg. A behavior model for persuasive design. In *Proceedings of the 4th International Conference on Persuasive Technology*, Persuasive '09, pages 40:1–40:7, New York, NY, USA, 2009. ACM. URL: http://doi.acm.org/10.1145/1541948.1541999, doi:10.1145/1541948.1541999.

[21] Shamsi T. Iqbal and Eric Horvitz. Notifications and awareness: A field study of alert usage and preferences. In *Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work*, CSCW '10, pages 27–30, New York, NY, USA, 2010. ACM. URL: http://doi.acm.org/10.1145/1718918.1718926, doi:10.1145/1718918.1718926.

[22] Thomas Jackson, Ray Dawson, and Darren Wilson. The cost of email interruption. *Journal of Systems and Information Technology*, 5(1):81–92, 2001. URL: http://dx.doi.org/10.1108/13287260180000760, arXiv:http://dx.doi.org/10.1108/13287260180000760, doi:10.1108/13287260180000760.

[23] Slack. Getting started with project management tools – wrike tour. URL: https://www.wrike.com/tour-wa/ [last accessed 2016-07-17].

[24] EPM Live. Epm live | online project management. URL: http://epmlive.com/ [last accessed 2016-07-16].

[25] EPM Live. Epm live | social collaboration. URL: http://epmlive.com/products/work-management/social-collaboration/ [last accessed 2016-07-16].

[26] Slack. Slack: Be less busy. URL: https://slack.com/is [last accessed 2016-07-18].

[27] Microsoft. Yammer features |team collaboration tools. URL: https://products.office.com/en-US/yammer/yammer-features [last accessed 2016-07-18].

[28] Microsoft. Yammer integration with office 365 - office 365. URL: https://support.office.com/en-us/article/Yammer-integration-with-Office-365-4086681f-6de1-4d39-aa72-752b2af1cbd7 [last accessed 2016-07-18].

[29] Telerik. Introduction | kendo ui docs. URL: http://docs.telerik.com/kendo-ui/introduction [last accessed 2017-01-19].

[30] jQuery. jquery api documentation. URL: http://docs.telerik.com/kendo-ui/introduction [last accessed 2017-01-22].